

NOAA Technical Report NOS CS 5

MECCA2 PROGRAM DOCUMENTATION

Silver Spring, Maryland
January 2000



noaa National Oceanic and Atmospheric Administration

U.S. DEPARTMENT OF COMMERCE
National Ocean Service
Coast Survey Development Laboratory

Office of Coast Survey
National Ocean Service
National Oceanic and Atmospheric Administration
U.S. Department of Commerce

The Office of Coast Survey (CS) is the Nation's only official chartmaker. As the oldest United States scientific organization, dating from 1807, this office has a long history. Today it promotes safe navigation by managing the National Oceanic and Atmospheric Administration's (NOAA) nautical chart and oceanographic data collection and information programs.

There are four components of CS:

The Coast Survey Development Laboratory develops new and efficient techniques to accomplish Coast Survey missions and to produce new and improved products and services for the maritime community and other coastal users.

The Marine Chart Division collects marine navigational data to construct and maintain nautical charts, Coast Pilots, and related marine products for the United States.

The Hydrographic Surveys Division directs programs for ship and shore-based hydrographic survey units and conducts general hydrographic survey operations.

The Navigation Services Division is the focal point for Coast Survey customer service activities, concentrating predominantly on charting issues, fast-response hydrographic surveys and Coast Pilot updates.

NOAA Technical Report NOS CS 5

MECCA2 PROGRAM DOCUMENTATION

Kurt Hess

January 2000



noaa National Oceanic and Atmospheric Administration

U.S. DEPARTMENT
OF COMMERCE
William Daly, Secretary

Office of Coast Survey
Captain David MacFarland

National Oceanic and
Atmospheric Administration
D. James Baker, Under Secretary

National Ocean Service
Nancy Foster
Assistant Administrator

Coast Survey Development
Laboratory
Bruce Parker

NOTICE

Mention of a commercial company or product does not constitute an endorsement by NOAA. Use for publicity or advertising purposes of information from this publication concerning proprietary products or the tests of such products is not authorized.

TABLE OF CONTENTS

LIST OF FIGURES	iv
1. INTRODUCTION	1
2. MODEL ATTRIBUTES	3
Numerical Code	3
Model Variables	3
Model Grid	4
Cell Attribute Codes	5
Atmospheric Heat Flux	7
Input and Output Files	7
3. IMPROVEMENTS	9
Corrections to Previous Version	9
Input Data Files	9
Ocean and River Boundaries	12
Vertical and Horizontal Diffusivity	13
Shallow Water Wind Stress Reduction	14
Internal-Mode Velocity Boundary Conditions	14
Density Function and Gradients	16
Output for Graphing	16
Non-linear Horizontal Advection	16
4. ACKNOWLEDGMENTS	18
5. REFERENCES	19
APPENDIX A. ERRATA SHEET	21
APPENDIX B. SAMPLE GEOGRAPHY FILE	23
APPENDIX C. SAMPLE CONTROL FILE	27
APPENDIX D. CODE LISTING	29

LIST OF FIGURES

Figure 2.1. Plan View and Isometric View of grid cells showing placement of variables.	4
Figure 2.2. Chesapeake Bay model grid showing water cells.	6
Figure 3.1. Wind vectors over the Chesapeake Bay grid	11
Figure 3.2. Instantaneous surface currents around Rattray Island, Australia..	17

1. INTRODUCTION

This report describes the latest version of the MECCA (Model for Estuarine and Coastal Circulation Assessment) code that was originally published by Hess (1989) and hereafter referred to as M1. The new version of the code was developed in response to demands for a hindcast model for Chesapeake Bay (Bosley, 1996; Bosley and Hess, 1998), and to a lesser extent, to provide a version which corrected some of the errors in the first release. The original code is still operable, provided a few corrections to the code are made as described in the errata sheet (Appendix A).

MECCA was originally developed to meet several shortcomings not found in existing models: (1) extensive documentation including step-by-step development of the applicable equations and their conversion into finite difference form; (2) creation of a standardized form for the input that requires little explanation, can be adapted to most regional applications, and minimizes recoding; (3) provision of a model with internal switches to selectively eliminate various terms in the equations for sensitivity purposes; and (4) the inclusion of imbedded, one-dimensional flow to better represent river and channel flows. This philosophy has not changed.

With the new version comes additional philosophic goals: (1) to provide a basic, streamlined (less code) version which requires the user to do more outside coding and (2) reading data from similar, external files to provide time series values for boundary conditions. For example, the new version reads all data files two records at a time; fewer values are stored at any time, but with linear interpolation the user needs to add more points in time to create a smooth curve - there is no cubic interpolation (which requires four time points). Other improvements include variable array sizes, corrections to the original code, and several other changes (see Section 3).

The model has been requested, distributed, and used numerous times. The author has completed several applications including the estimation of Chesapeake Bay's natural period (Hess, 1988a), sediment transport (Hess, 1988b), and crab larval drift (Johnson and Hess, 1990). It has been used for the Gulf of Maine (Brooks, 1992; Brooks and Churchill, 1992; Brooks, 1994) and in Australian coastal waters (Galloway et al., 1996 - but see Section 4; King et al., 1998). It has been applied to coastal flows in France (Smaoui, 1996; Berthet, 1996).

This report is intended to describe the new features with a short review of the previous version (see M1 for details). The report covers a brief overview of the model in Section 2, a description of new features in Section 3, and an overview of applications to Chesapeake Bay and Rattray Island, Australia, in Section 4. Appendix A contains an errata sheet for the original version of MECCA. Appendix B has a new sample Geography File and Appendix C has a new sample Control File. Appendix D contains a listing of the MECCA2 code.

2. MODEL ATTRIBUTES

The following is a brief discussion of MECCA attributes as discussed in M1 which serves as a refresher in preparation to the description of the modifications in Section 3.

Numerical Code

The MECCA code solves the hydrodynamic equations of momentum, mass, salinity, and temperature conservation. It is three-dimensional in space, uses a vertical sigma coordinate, has a time-varying free surface, and incorporates non-linear horizontal momentum advection. It includes a three-dimension time variable horizontal diffusion based on Smagorinsky (see Tag et al., 1979) and includes vertical turbulent diffusion based on a mixing length and Richardson number-dependent reduction (Munk and Anderson, 1948). For the horizontal momentum equations, the external gravity wave mode is split apart from the internal mode.

Variables are placed on an Arakawa C-grid with square cells in the horizontal and at uniform intervals along a sigma-stretched vertical coordinate. External-mode momentum is solved with an alternating-direction, semi-implicit method in the horizontal. The salinities, temperatures, and internal-mode velocities are solved with a semi-implicit method in the vertical.

The sigma vertical coordinate is defined here as

$$\sigma = \frac{z - \eta}{d - \eta} \quad (2.1)$$

where η is water level departure from the reference surface ($z = 0$) and d is the depth relative to the reference surface. In recent years, some modelers have encountered certain problems with sigma coordinate systems (Haney, 1991). These problems arose from accurately representing the horizontal pressure gradient due to density, and can be overcome by using uniformly-spaced sigma levels and by subtracting the spatially-averaged density before computing the horizontal gradient. MECCA has both these features.

The model is coded in Fortran with a modest amount of vectorization. Constants are read in from a Control File, and basin attributes are stored in a Geography File. Output is saved at the end of the run to provide a restart capability.

Model Variables

Two-dimensional variables include mean sea level depth (D), water level (SE, SEP, SEPP), vertically-integrated velocities (UH, UHP, VH, VHP), bottom stress (TBX, TBY), wind (WX, WY), surface stress (TSX, TSY), vertically-integrated horizontal turbulent viscosity (AH), and velocity departure functions (THETA1, THETA2, THETA3, THETSU, THETSV), cell status (IFIELD), time-integrated variables (SOLD, UHOLD, VHOLD), channel width (BX, BY), flow indices (MFLUX, NFLUX), edge parameter (FEDGE), relative cell area (AREA), vertically-averaged horizontal diffusivity (AH and AHC), and imbedded channel widths (BX, BY).

Three-dimensional variables include internal mode velocities (U , V , W), vertical viscosity (AV), vertical diffusivity (DV), salinity (S), temperature (T), Richardson Number (R), and horizontal viscosity ($AH3$). All units are metric unless otherwise stated.

Model Grid

Variables in the numerical grid are indexed by M in the x direction, N in the y direction and L in the $-q$ direction. Placement of variables in grid cells are shown in Figure 2.1.

The positions of cell boundaries in the grid's horizontal plane determined by

$$x = N\Delta \text{ and } y = M\Delta, \quad (2.2)$$

where Δ is the grid size in meters. Position in the vertical is determined by

$$z = (1-L)\delta, \quad (2.3)$$

where $\delta = 1/(LBOT - 1)$ and LBOT is the number of the level that correspond to the bottom.

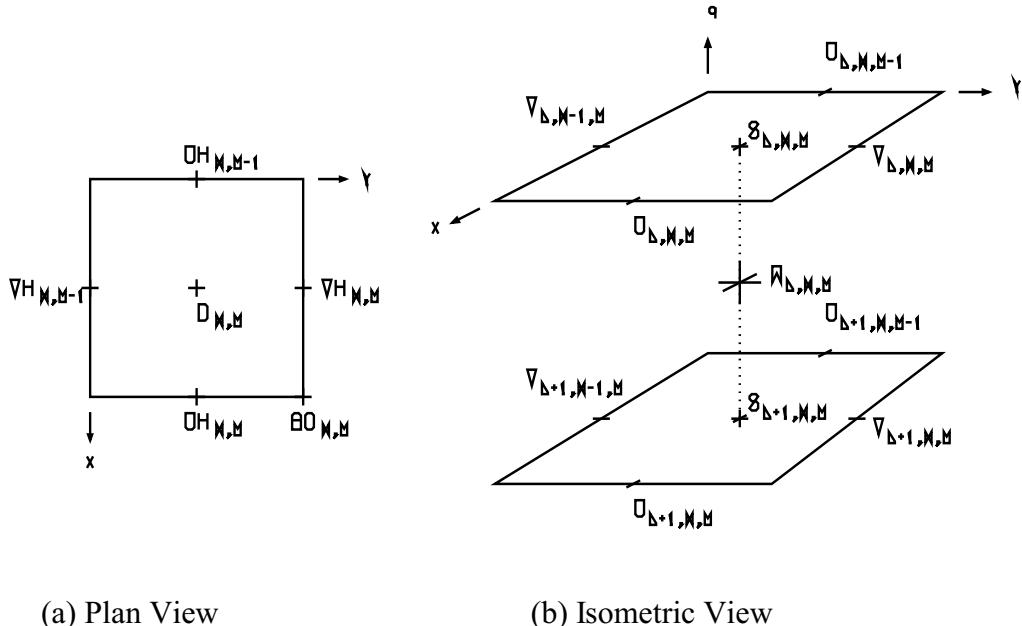


Figure 2.1. Plan View (a) and Isometric View (b) of grid cells showing placement of variables. The Plan View shows the two-dimensional variables. At D there are also SE, AH, FEDGE, AREA, PHI, WX, WY, IFIELD, FX, FY, TSX, TSY, and SOLD. At UH there are also U , UHOLD, UE, MFLUX, TBX, BX, GSTARX, THETA1, THETSU. At VH there are also V , VHOLD, VE, NFLUX, TBY, BY, GSTARY, THETA2, and THETSV. At AC (which is AHC) there is also THETA3. The Isometric View (b) shows the location of three-dimensional variables. At U there is also GRX and at V there is also GRY. At S there is also T and $AH3$. At W there is also AV and DV .

The grid is oriented to the surface of the earth by the following variables in the Geography File (Appendix B): BSNANG, BSNLAT, BSNLON, NCOR, and MCOR. The lower corner (i.e., closest to the origin) of cell ($n = \text{NCOR}$, $m = \text{MCOR}$) is at latitude BSNLAT and longitude BSNLON. The lower corner corresponds to location $x = (\text{NCOR} - 1)\Delta$ and $y = (\text{MCOR} - 1)\Delta$, where Δ is the grid size. The y axis is oriented at an angle BSNANG clockwise from due east. Suppose μ_x is the Mercator transform from degrees longitude to X and μ_y is the Mercator transform from degrees latitude to Y. Then the Mercator coordinates of any cell's lower corner are

$$\begin{aligned} X &= \mu_x\{\text{BSNLON}\} - (M - MCOR) \Delta' \sin\{\text{BSNANG}\} + (N - NCOR) \Delta' \cos\{\text{BSNANG}\} \\ Y &= \mu_y\{\text{BSNLAT}\} - (M - MCOR) \Delta' \cos\{\text{BSNANG}\} - (N - NCOR) \Delta' \sin\{\text{BSNANG}\} \end{aligned} \quad (2.4)$$

where

$$\Delta' = \mu_y\{\text{BSNLAT} + \frac{\Delta}{2}\} - \mu_y\{\text{BSNLAT} - \frac{\Delta}{2}\} \quad (2.5)$$

and Δ° is the grid size converted to degrees of latitude

$$\Delta^\circ = \frac{\Delta}{1852 \times 60} \quad (2.6)$$

The grid for Chesapeake Bay is shown in Figure 4.2 (Bosley and Hess, 1998; Bosley, 1996). Simulations are being made on a grid with bathymetry that was previously developed at the U.S. Naval Academy (Hoff, 1990). The grid cell size (Δ) is 5,606 m and the model was run in barotropic mode.

Cell Attribute Codes

Each cell in the Geography File (Appendix B) is tagged with a two-digit number that defines certain attributes. The entire status of the cell is stored in the two-dimensional array IFIELD, where

$$\text{IFIELD} = 10(I) + J \quad (2.7)$$

For example, a cell may represent either land or water, and if water it may be an ocean or river boundary cell. It may also be either a full or half (triangular) cell, although boundary cells must be full. This is the cell's geographic status, and is coded in the single-digit integer I. For example,

I =	1 denotes a triangle
	2 denotes a full water cell
	5 denotes an ocean boundary cell
	6 denotes a river boundary cell

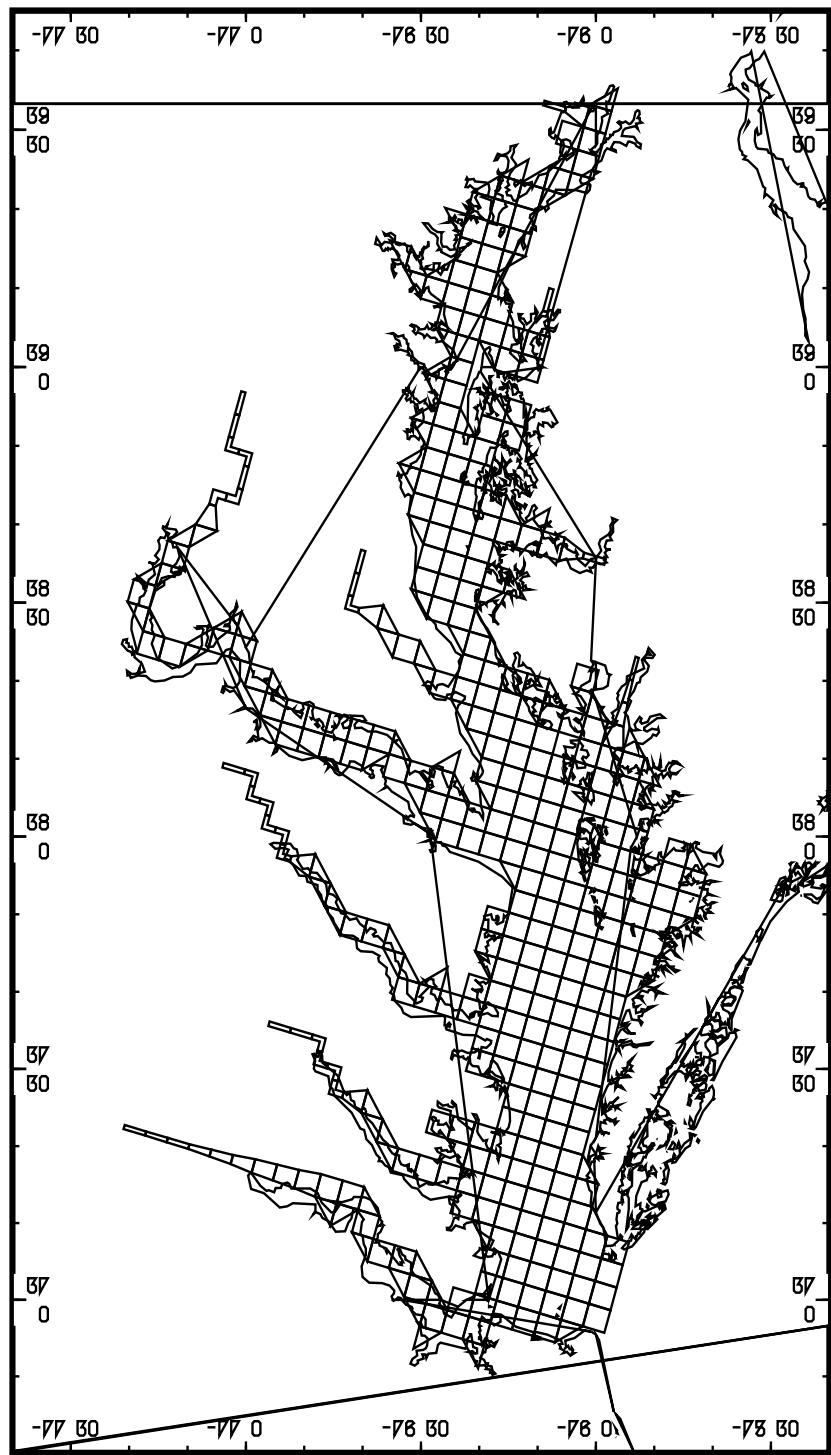


Figure 2.2. Chesapeake Bay model grid showing water cells. Cells are 5606 m on a side. Upper reaches of rivers are modeled as narrow channels.

A second cell feature is the potential existence of a physical barrier that prevents flow in either or both directions, and is coded in the single-digit integer J. For example,

J =	0 denotes no barriers
	1 denotes a barrier in x-direction
	2 denotes a barrier in y-direction
	3 denotes barriers in both the x- and y-directions

The program uses IFIELD primarily to skip over land cells and to enforce zero flux at barriers.

NOTE: because MECCA sometimes makes calculations at cells near the edge of the grid, it is desirable to have extra rows and columns of non-water cells at the outer boundaries of the grid.

Atmospheric Heat Flux

The atmospheric heat flux formulation in M1 was tested with data for Tampa Bay (Hess, 1994) and found to be satisfactory. There are only a few minor changes to the code. Bottom heat flux has been removed since it had little effect on computed water temperatures.

Input and Output Files

MECCA is tailored for a specific application by reading data from a set of input files. Time step, printing, and other data, as well as the names of all other input files, are read in from the Control File. The grid, depths, orientation, and cell size are read from the Geography File. Environmental, or driving, data are read from additional files (see Section 3). If the run is restarted from a previous run, an Initialization File is read.

MECCA output is put into another set of files. These are the Screen File, which lists current timestep, the Print File, which contains top views and side views of various variables, and the Graphing File, which contains date and time series values at selected locations. A Save File is created which can be used for an Initialization File.

3. IMPROVEMENTS

Corrections to Previous Version

The previous version of the code (M1) is still very usable, provided a few corrections, especially one to the non-linear calculations in the internal-mode module, are made. A table showing the suggested changes appears in Appendix A.

Input Data Files

The boundary condition data are now read from separate input files, not from the Control File. In addition, to reduce array storage requirements, MECCA now stores only two records of data at a time; values needed in the program are based on a linear interpolation between the two values read in.

The following are, in the order they are read in the .CON file, the seven input data file types that can be read in MECCA:

Ocean Water Levels or Flowrates
Winds or Stresses and (Optional) Pressures Gradients
River Flowrates
Ocean Salinities
Ocean Temperatures
River Temperatures
Additional Meteorological Data

In the .CON file, for each of the above seven types, MECCA reads (1) a text file description (not used in computation), (2) the number of signals (NSIGS) in the file to be read, and (3) the file name. If NSIGS is 0, the file name is not read and therefore no data are read (all array values for that variable are zero).

For all files (except the wind/stress file), each record has a fixed format. A typical record contains the four-digit year, the day in year, and a number (=NSIGS) of values. The form is

YYYY DDD.DDDD V1 V2 V2 V3 V5

where YYYY is the 4-digit year, DDD.DDDD is a day-in-year date, and V1, V2, etc. are a set of values corresponding to that time given. Spaces or commas should separate all numbers. For example, a typical Additional Meteorological Data file is

1994	3.5000	12.04	0.50	0.32	1013.80
1994	3.6250	13.25	0.56	0.44	1014.10
1994	3.7500	12.72	0.63	0.30	1014.30
1994	3.8750	12.03	0.61	0.22	1014.20
1994	4.0000	10.86	0.59	0.29	1014.20

which contains the year, day, air temperature, relative humidity, cloud cover, and air pressure.

The time values are converted to a year-based time, YT, using the universal time coordinate (UT) and the number of days in the year, D. Note that noon on January 1 corresponds to UT = 1.5.

$$YT = (YEAR - 1900) + \frac{UT - 1}{D} \quad (3.1)$$

Wind data, unlike the other data types which are in ASCII, are stored in a binary file. These data require two unformatted records per time which contain:

YEAR, UTC, ITYPE1, ITYPE2, ITYPE3, NX, MX
FX, FY, (DPADX, DPADY)

Here YEAR and UTC are the date stamp; ITYPE1, ITYPE2, and ITYPE3 are indices (see below); and NX and MX are array sizes. The arrays FX(NX, MX) and FY(NX, MX) contain either wind speeds or wind stresses (per unit water density). DPADX and DPADY are atmospheric pressure gradients (mb/km). The indices are as follows: for FX and FY containing winds, ITYPE1 = 1; for FX and FY containing stresses, ITYPE1 = 2. For no atmospheric pressure gradient values to be read, ITYPE2 = 0; for values to be read, ITYPE2 = 1. For ITYPE3 = 0, model coordinate directions are used for winds, stresses, and pressure gradients. For ITYPE3 = 1, earth coordinate directions (east, north) are used.

For a wind with eastward and northward components WE and WN respectively, the components can be converted to model directions by

$$WX = -WN\cos\{BSNANG\} - WE\sin\{BSNANG\} \quad (3.2a)$$

$$WY = WE\cos\{BSNANG\} - WN\sin\{BSNANG\} \quad (3.2b)$$

The components can be converted back to earth coordinates by

$$WE = -WX\sin\{BSNANG\} + WY\cos\{BSNANG\} \quad (3.3a)$$

$$WN = -WX\cos\{BSNANG\} - WY\sin\{BSNANG\} \quad (3.3b)$$

A sample wind field for Chesapeake Bay is shown in Figure 3.1.

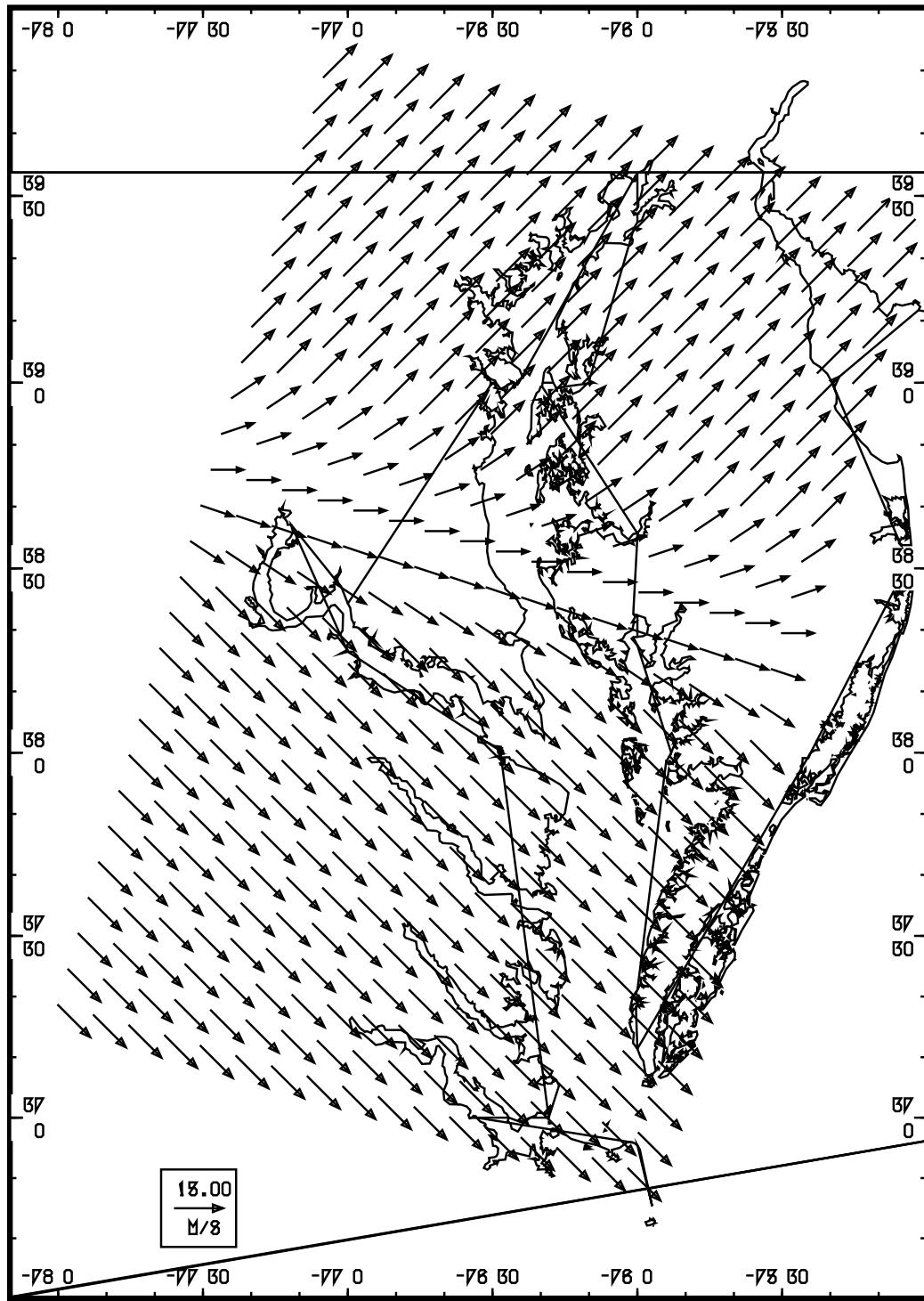


Figure 3.1. Wind vectors over the Chesapeake Bay grid. The wind vectors generated by applying the spatial interpolation scheme to a northeastward wind of 14 m/s at Thomas Point and a southeastward wind of 14 m/s at CGBT. Vectors at every other cell are plotted.

Ocean and River Boundaries

Ocean and river boundaries are defined in a new way. For an ocean boundary, as specified in the Geography File (Appendix B), the indices are

MB1, MB2, NB1, NB2, ITPO, JTPO, ISET1, ISET2

For an ocean boundary, the cells located at $MB1 \leq M \leq MB2$ and $NB1 \leq N \leq NB2$. The direction of outflow is set by ITPO, where

ITPO =	1 denotes flow in +x direction -1 denotes flow in -x direction 2 denotes flow in +y direction -2 denotes flow in -y direction
--------	--

The type is set in JTPO, where

JTPO =	1 denotes a water level 2 denotes a flowrate per unit width. 3 denotes an Orlanski radiation condition 4 denotes a Riemann invariant condition
--------	---

The Orlanski (1976) radiation condition is

$$\frac{\partial \eta}{\partial t} + c \frac{\partial \eta}{\partial x_n} = 0 \quad (3.4)$$

and the Riemann invariant condition (Wurtele et al., 1971) is

$$\eta + \frac{H}{c} (u^2 + v^2)^{1/2} = 0 \quad (3.5)$$

where c is the shallow water wave speed, H is total water depth, and x_n is the outward normal direction.

ISET1 and ISET2 are ocean signal numbers; water level at any cell is linearly interpolated in space between the present (time-interpolated) value in signal ISET1 at (NB1, MB1) and the value in signal ISET2 at (NB2, MB2).

For a river boundary, the indices are

MR1, MR2, NR1, NR2, ITPR, JTPR, ISETR

The direction of inflow, ITPR, is set analogously to ITPO, except that for example +1 means inflow in the +x direction (not outflow). The type is set in JTPR, where

JTPR = 1 denotes simple flowrate
 2 denotes water falls

ISETR is river signal number, so that the first river so defined used signal ISETR in the river flowrate and temperature files. Also, there is a revised water falls condition. Unlike before (see M1, p. II-12 to II-15), the boundary water level increment and temperature are introduced directly into the river cell, not into the adjacent cell.

Ocean boundaries may have one or more of the following variables assigned as a boundary condition: water level, salinity, and temperature. River boundaries may have one or more of the following variables assigned as a boundary condition: flowrate, salinity, and temperature.

Vertical and Horizontal Diffusivity

Vertical mixing is handled in a more complex way and allows for negative Richardson Number (R) values. Negative values occur when the density is unstably stratified, and the new scheme causes large vertical diffusivities. The approach is based on that of Munk and Anderson (1948). As before, viscosity and diffusivity are related to a potential, A_z as follows:

$$A_z = \left(\kappa H q^{C_0} (1 + q) \right)^2 \left[\left(\frac{\partial u}{\partial z} \right)^2 + \left(\frac{\partial v}{\partial z} \right)^2 \right]^{1/2} \quad (3.6)$$

$$A_v = A_{vo} + f_A A_z \quad (3.7)$$

$$D_v = D_{vo} + f_D A_z \quad (3.8)$$

where κ is von Karman's constant, A_{vo} and D_{vo} are small (molecular scale) values, and the functions f_A and f_D adjust for stratification.

Negative values of R, which imply denser water lying above lighter water, now create augmented mixing. For vertical momentum viscosity

$$\begin{aligned} f_A &= C_1 (1 + C_2 R)^{C_3} && \text{for } R \geq 0 \\ &= C_1 (1 + C_4 R^2) && \text{for } R < 0 \end{aligned} \quad (3.9)$$

And for vertical diffusivity,

$$\begin{aligned} f_D &= C_5 (1 + C_6 R)^{C_7} && \text{for } R \geq 0 \\ &= C_5 (1 + C_8 R^2) && \text{for } R < 0 \end{aligned} \quad (3.10)$$

Also, the values of R are bounded:

$$R_{\min} < R \leq R_{\max} \quad (3.11)$$

The 13 mixing variables (A_{vo} , D_{vo} , R_{\min} , R_{\max} , $C_0 - C_8$) are coded as (AV0, DV0, RIMIN, RIMAX, CR0, CRICH(1) - CRICH(8)) and have the nominal values (0.00001, 0.000005, -0.05, 1000., 1.0, 0.4, 10.0, 0.5, 20000., 0.1, 3.33, 1.5, 20000.). There is now no subroutine MIXUP, which (in M1) created a neutral density distribution.

Horizontal diffusivity has been improved by making it three dimensional. Horizontal diffusivity is the product of the input parameter DHDA and the viscosity, AH3.

Shallow Water Wind Stress Reduction

Wind stress is reduced in shallow water to avoid high velocities. In the code, wind stress is now multiplied by the factor f_τ , where

$$\begin{aligned} f_\tau &= 1 & d_2 &\leq H \\ &= \frac{H - d_1}{d_2 - d_1} & d_1 &< H \leq d_2 \\ &= 0 & H &< d_1 \end{aligned} \quad (3.12)$$

and H is the total water depth. The values (d_1 , d_2) are coded as (DTAU1, DTAU2) and have values (0.10, 1.0).

Internal-Mode Velocity Boundary Conditions

There is a new upper boundary condition on internal-mode velocity. At the upper surface, a one-sided approximation may be used by setting ITOPV=3. In that case, horizontal velocity is computed by setting the upper surface's vertical diffusion and advection to zero and reducing volume-related terms by half (as is done for the upper salinity and temperature boundary conditions: see M1, p I-77). In the new version of MECCA, the bottom stress per unit density is defined as the product of a friction factor and a representative velocity by the generalized form

$$\frac{\tau_b}{\rho} = \Phi [U + \gamma u_{LBOT} + (1 - \gamma) u_{LBOT-1}] \quad (3.13)$$

where Φ is the friction factor, U is the external-mode velocity, and u is the internal-mode velocity. For the general case,

$$\Phi = C_{DWB1} + C_{DWB2} |U + u_{LBOT}| \quad (3.14)$$

and $\gamma = 1$.

A new feature is the specification of a bottom logarithmic boundary layer with velocity

$$u(z) = \frac{1}{\kappa} \left(\frac{\tau_b}{\rho} \right)^{1/2} \ln \left(\frac{z}{z_o} \right) \quad (3.15)$$

The roughness height z_o is defined in the bottom stress subroutine as 0.003 m. At mid-depth in the bottom layer (which is at a distance of $H\delta/2$ above the bottom), we set the boundary layer velocity to be equal to the mean of the lowest two sigma level velocities. Squaring each side and rearranging gives

$$\frac{\tau_b}{\rho} = \left(\frac{\kappa}{\ln \left\{ \frac{H\delta}{2z_o} \right\}} \right)^2 (U + \frac{1}{2} u_{LBOT} + \frac{1}{2} u_{LBOT-1})^2 \quad (3.16)$$

Therefore, Φ is determined to be

$$\Phi = \left(\frac{\kappa}{\ln \left\{ \frac{H\delta}{2z_o} \right\}} \right)^2 |U + \frac{1}{2} u_{LBOT} + \frac{1}{2} u_{LBOT-1}| \quad (3.17)$$

and $\gamma = 0.5$. The log-layer bottom boundary condition is activated by setting IBOTV=3.

Another bottom condition, which was included in the older version of MECCA, is one of no slip, or

$$U + u_{LBOT} = 0. \quad (3.18)$$

In this case, the bottom stress is set equal to the product of the viscosity at the mid-level of the lowest layer and the velocity gradient

$$\begin{aligned} \frac{\tau_b}{\rho} &= \frac{A_v}{H\delta} \left((U + u_{LBOT-1}) - (U + u_{LBOT}) \right) \\ &= \frac{A_v}{H\delta} (U + u_{LBOT-1}) \end{aligned} \quad (3.19)$$

In the new version, the above expression is recast in the form of (3.13) by setting

$$\Phi = \frac{A_v}{H\delta} \quad (3.20)$$

and $\gamma = 0$. The no slip bottom boundary condition is activated by setting IBOTV=0.

Density Function and Gradients

Although there is a function for water density, there is no longer a separate function for difference in density. As before,

$$\rho = 10^3 (1 + FRHO\{S, T\}) \quad (3.21)$$

The density difference is now computed directly by the difference in values of FRHO. This allows other density formulations to be used.

When either salinity or temperature are not being updated (c.f., KONCEN), they are set to their respective constant default values SAL0 (30 ppt) and TMP0 (10 C). If either salinity or temperature are being updated and a boundary file is missing, boundary values are set to the respective default values. This allows the full density equation (3.21) to be used with representative salinity or temperature values.

Vertical and horizontal density gradients can now be excluded selectively. For ICOUPL=0, both the vertical and the horizontal gradients are set to zero. For ICOUPL=1, only vertical gradients are included. For ICOUPL=2, both vertical and horizontal gradients are included.

Output for Graphing

The output file containing time series of model values at successive times has a new format. The first value in each record is the year, the second is the Julian day, and subsequent values are the selected modeled values. This makes them identical to the input data files.

Other attributes of the graphical file are selected by the parameters IGPH, NSTGPH, and IGPHOP (see Appendix C, Sample Control file). IGPH is the number of variables (as denoted by location L, N, M and ITYP) to be saved (up to 20). NSTGPH is the interval, in internal-mode time steps, that data are to be saved. IGHPOP (when set to 0) causes the graphical file header information not to be printed.

Non-Linear Horizontal Advection

The new code contains the corrected versions of the non-linear horizontal momentum advection terms in the internal mode calculations. For the terms affected, see Appendix A. These terms are approximated by upstream differencing as follows:

$$\frac{\partial uu}{\partial x} \rightarrow \frac{1}{\Delta} \left[\left(1 - \frac{u_m}{|u_m|}\right) (u_{m+1}^2 - u_m^2) + \left(1 + \frac{u_m}{|u_m|}\right) (u_m^2 - u_{m-1}^2) \right] \quad (3.22)$$

The original code was corrected in about 1990 although the older version apparently did not affect

the Gulf of Maine simulations (Brooks, 1992; Brooks and Churchill, 1992; Brooks, 1994). However, MECCA was used for a model inter-comparison study (Galloway et al., 1996). They simulated tidal flow around Rattray Island on Australia's east coast in an attempt to reproduce lee eddies that have been observed. Unfortunately, the authors were not successful in producing eddying flow with MECCA because they did not obtain a corrected version of the model. Although the authors were given a updated version of the code and plots showing the lee eddies, this information arrived too late to in their paper. Nevertheless, MECCA did produce eddies (Figure 3.2) that are virtually identical to those produced by the other three-dimensional models.

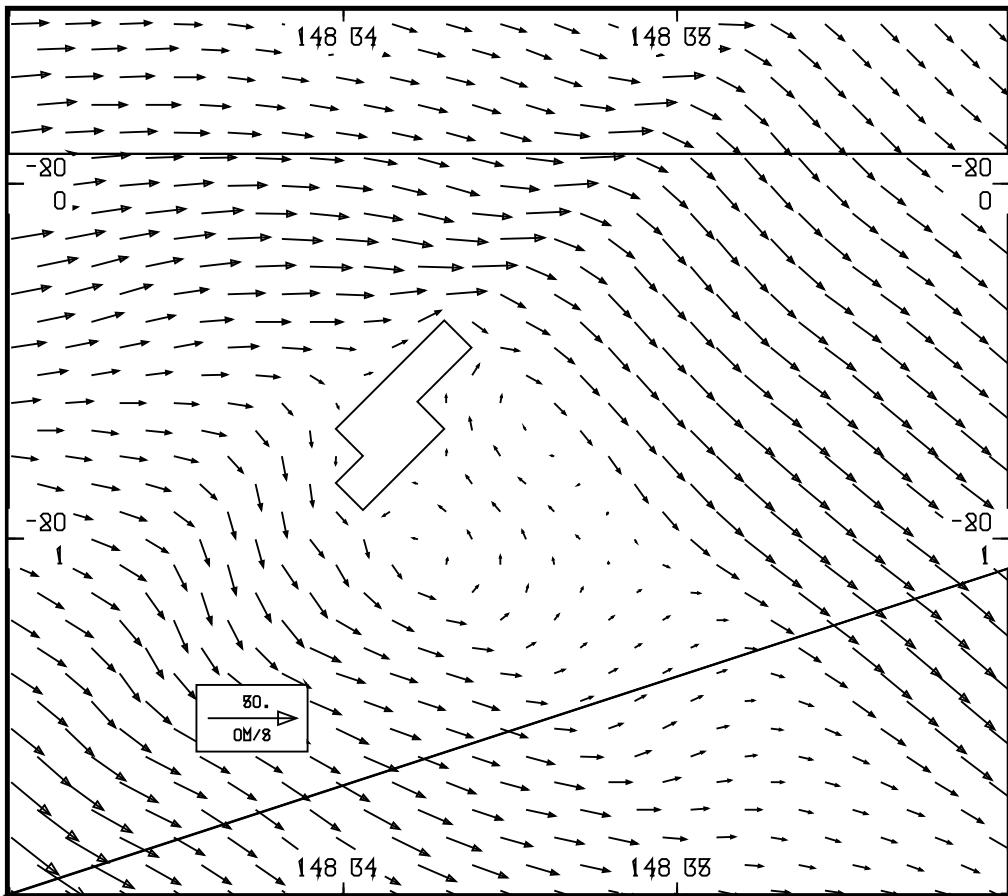


Figure 3.2. Instantaneous surface currents around Rattray Island, Australia.

4. ACKNOWLEDGMENTS

The author is indebted to Dr. Thor Aarup, who has been instrumental in modernizing portions of the code to make it more flexible. He has also assisted greatly in disseminating the code to other interested parties. Dr. Hassan Smaoui diligently checked the code and provided key corrections in the velocity calculations. He has also extended the model's capabilities by introducing a turbulence closure scheme for the diffusivity. Dr. Kathryn Bosley has also greatly spurred development of the new version by creating the specific needs of the Chesapeake Area Forecasting Experiment, in testing the model in a variety of scenarios, and in critically assessing the model's performance and assumptions.

5. REFERENCES

- Berhet, C., 1996: Flow and three-dimensional coastal transport: numerical applications. Ph.D. thesis, Universite de Joseph Fourier, Grenoble, France (in French).
- Bosley, K. T., and K. Hess, 1998: Development of a Experimental Nowcast/forecast System for Chesapeake Bay Water Levels. **Proceedings, 4th International Conference on Estuarine and Coastal Modeling**, Alexandria, VA, October 20-22, 1996. 413 - 426.
- Bosley, K. T, 1996: Research aimed at prediction of water levels in the Chesapeake Bay. **Proceedings, Conference on Coastal Oceanic and Atmospheric Prediction**. AMS, Atlanta, 268 - 271.
- Brooks, D. A., 1994: A model study of the buoyancy-driven circulation in the Gulf of Maine, **Journal of Physical Oceanography** 24 (11), 2387-2412.
- Brooks, D. A., 1992: Tides and tidal power in Passamaquoddy Bay: a numerical simulation. **Continental Shelf Research** 12, 675-716.
- Brooks, D. A., and L. U. Churchill, 1992: Experiments with a terrain-following hydrodynamic model for Cobscook Bay in the Gulf of Maine. In: **Estuarine and Coastal Modeling** (Spaulding et al., Eds), Am. Soc. Civil Eng., 786 pp.
- Galloway, D., E. Wolanski, and B. King, 1996: Modeling eddy formation in coastal waters: a comparison between modeling capabilities. **Proceedings, 3rd International Conference on Estuarine and Coastal Modeling**, 13 - 25.
- Haney, R. L., 1991: On the pressure gradient force over steep topography in sigma coordinate ocean models. **Journal of Physical Oceanography**, 21, 610 - 619.
- Hess, K. W., 1994: Tampa Bay Oceanography Project: Development and Application of the Numerical Circulation Model. U.S. Department of Commerce, **NOAA Technical Report NOS OES 005**. 90 pp.
- _____, 1989: MECCA Program Documentation. U.S. Department of Commerce, **NOAA Technical Report NESDIS 46**. 258 pp.
- _____, 1988a: Linearized numerical model of constituent tides in Chesapeake Bay.(unpublished manuscript)
- _____, 1988b: Lagrangian drift model of suspended sediment transport in Chesapeake Bay, **Proceedings, Understanding the Estuary: Advances in Chesapeake Bay Research**, 352-368.

Hoff, M., 1990: A Chesapeake Bay Circulation Model. US Naval Academy, Annapolis, 26 pp.

Johnson, D. F., and K. W. Hess, 1990: Numerical simulations of blue crab larval dispersal and recruitment, **Bulletin of Marine Science**, 46(1), 195-213.

Johnson, D. F., and K. W. Hess , 1990: Numerical simulations of blue crab larval dispersal and recruitment, **Bulletin of Marine Science**, 46(1), 195-213.

King, B., S. Spagnol, E. Wolanski, and T. Done, 1997: Modeling the Mighty Burdekin River in Flood. **Proceedings, 4th International Conference on Estuarine and Coastal Modeling**, Alexandria, VA, October 20-22, 1996. 103-115.

Munk, W. H., and E. R. Anderson, 1948: Notes on the theory of the thermocline. **Journal of Marine Research**, 7, 276 - 295.

Smaoui, H., 1996: Three-dimensional numerical modeling of hydrodynamics and sediment transport in the eastern part of the English Channel and the southern part of the North Sea. Ph.D. thesis, University of Lille, France (in French).

Tag, P. M., F. W. Murray, and L. R. Koenig, 1979: A comparison of several forms of eddy viscosity parameterization in a two-dimensional cloud model. **Journal of Applied Meteorology**, 18, 1429 - 1441.

Wu, J., 1980: Wind-stress coefficients over sea surface under neutral conditions - a revisit. **Journal of Physical Oceanography**, 10, 727 - 740.

Wurtele, M. G., J. Paegle, and A. Sielecki, 1971: The use of open boundary conditions with the storm surge equations. **Monthly Weather Review**, 99, 6, 537 - 544.

APPENDIX A. ERRATA SHEET

This section contains some of the errors in the MECCA NESDIS report (Hess, 1989) in both the text (Part I) and the code (Part III).

Text Errors

No.	Page	Comments
-----	------	----------

1. I-20 In Eq. 3.34, there should be no negative sign.

2. I-24 In Eq. 4.16, Θ_{sv} should be $H\Theta_{sv}$

3. I-33 Add the definition for $F_{n,m}$ (Eq. 4.8)

$$F_{n,m} = (\Phi_{n,m+1} + \Phi_{n,m}) / (2H_{n,m}) + \beta_c THETSU_{n,m}$$

where $\Phi_{n,m}$ is evaluated at the cell center by Eq. 4.7 and
LBOT-1

$$\begin{aligned} THETSU_{n,m} = & C_{drgws} | UE_{n,m} | / (\Delta L BX_{n,m}) \sum_{k=1}^{LBOT-1} [CI_k (1 \\ & + U_{n,m,k} / \underline{UE}_{n,m}) | (1 + U_{n,m,k} / \underline{UE}_{n,m}) |] \end{aligned}$$

And $\underline{UE}_{n,m} = \text{sign}(UE_{n,m}) \max(|UE_{n,m}|, 0.001)$

and $CI_k = 1/(LBOT - 1)$

$$\text{Add } C_{4A} = 2\Delta T / \Delta L^2$$

Then in Eqs. 5.28 and 5.29, C_4 should be replaced by C_{4A} .

4. I-42 Add the definition for $F_{n,m}$

$$F_{n,m} = (\Phi_{n+1,m} + \Phi_{n,m}) / (2H_{n,m}) + \beta_c THETSV_{n,m}$$

where

$$\begin{aligned} THETSV_{n,m} = & C_{drgws} | VE_{n,m} | / (\Delta L BY_{n,m}) \sum_{k=1}^{LBOT-1} [CI_k (1 \\ & + V_{n,m,k} / \underline{VE}_{n,m}) | (1 + V_{n,m,k} / \underline{VE}_{n,m}) |] \end{aligned}$$

$$+ V_{n,m,k} / \underline{VE}_{n,m}) | (1 + V_{n,m,k} / \underline{VE}_{n,m}) |]$$

And $\underline{VE}_{n,m} = \text{sign}(VE_{n,m}) \max(|VE_{n,m}|, 0.001)$

5. I-43 In Eqs. 6.20 and 6.21, C_4 should be replaced by C_{4A} .
6. I-54 In Eq. 8.33, the term $BHI_{n,m}$ (in first line) should be multiplied by 2.
Therefore, Eq. 8.34 should be revised to read: $C_3 = \Delta T / \Delta L^2$ (also applies to p. I-64, but text reads correctly)
7. I-55 In Eq. 8.45, the term multiplying β_a (first line) should be multiplied by 4.
Therefore, add a new variable: $C_{7A} = \Delta T / (4\Delta L)$
8. I-56 In first line of Eq. 8.49, replace C_7 by C_{7A} .
9. I-63 In Eq. 9.3, Θ_{sv} should be $H\Theta_{sv}$
10. I-65 In first line of Eq. 9.26, replace C_7 by C_{7A} .
-

Program Errors

No.	Page	Comments
1.	III-70	Line 325, add $CX9=2.*DT/DL**2$. Lines 393, 394: change CX4 to CX9.
	III-70	Line 357, change 10. to 0.1/RHOW
	III-71	Line 413, the variable should be AHDUYY
	III-72	Line 477, change 10. to 0.1/RHOW
	III-73	Lines 513 and 515: change CX4 to CX9.
2.		Non-linear terms in Subroutine UPVP
	III-80	After Line 312, add $B8=DTT/(4.*DL)$
	III-81	Lines 388 and 389, change B7 to B8. Line 410, change UPM(L,M) to UPM(L,N)
	III-83	Lines 509 and 510, change =HI.. to =B8*HI.. Line 540, change VPMM to VPM.

APPENDIX B. SAMPLE GEOGRAPHY FILE

Sample MECCA2 Geography File.

26	16	0.20	1.00	0.20	
25	28	1.00	0.15	0.15	(NANTICOKE)
26	28	1.00	0.15	0.15	
27	28	1.00	0.15	0.15	
34	12	0.20	1.00	0.20	(RAPPAHANNOCK)
34	13	0.20	0.20	0.20	
35	13	0.20	0.20	0.20	
35	14	0.20	0.20	0.20	
36	14	0.22	0.22	0.22	
36	15	0.22	0.22	0.22	
37	15	0.24	0.24	0.24	
37	16	0.25	1.00	0.25	
44	17	0.20	1.00	0.20	(YORK)
44	18	0.20	1.00	0.20	
44	19	0.22	0.22	0.22	
45	19	0.22	0.22	0.22	
45	20	0.24	0.24	0.24	
46	20	0.25	0.25	0.25	
50	12	0.18	1.00	0.18	(JAMES)
50	13	0.19	1.00	0.19	
50	14	0.20	1.00	0.20	
50	15	0.21	1.00	0.21	
50	16	0.22	1.00	0.22	
50	17	0.23	1.00	0.23	
50	18	0.24	1.00	0.24	
50	19	0.25	1.00	0.25	

----- END OF FILE -----

APPENDIX C. SAMPLE CONTROL FILE

MECCA2 Control File. Asterisks in rightmost column denote lines that differ from the original version.

```

MECCA VERSION 2.0                               Public Release Version
fullbay20.geo                                  NAME OF GEOGRAPHY FILE
0 1 1                                         FILE VERSION, ITEST, KTEST      *
----- MODEL CONFIGURATION PARAMETERS -----
RUN PARAMETERS
240.0 24.00 0.0 24.                         HRMAX, HROUT, HROUT0, HRSAVE      *
TIMESTEP(external), SPLIT, NUMBER OF LAYERS
225.00 4 9                                     DTE, ISPLIT, LAYRS
TURBULENCE VARIABLES
1.0 1.0 .01 1.0 -.05 1.E+3                  AH00,AH0,CAH,DHAH,RIMIN,RIMAX*
.003 .000010 .40 10.00 0.5 2.E+4            AV00,AV0,CRICH(1 - 4)          *
.001 .000005 .10 3.33 1.5 2.E+4            DV00,DV0,CRICH(5 - 8)          *
1 1 1.0                                         IHVISC,IVISC,CRO              *
DRAG COEFFICIENTS
.0007 .0000 .000 .0008 .000065             CDWB1,CDWB2,CDRWS,CDRG1,CDRG2
HEATING CONSTANTS
0.10 6.0                                       ALB,D10PCT                   *
SWITCHES
1 1 1 1                                         ICOR,IBETAA,IBETAP,IBETAH
1 0 0 0                                         IEXTRN,INTER,KONCEN,ICOUPL
3 2 0 0                                         ITOPV,IBOTV,IHEAT,ICPOS      *
----- PRINT PARAMETERS -----
PLAN VIEW VARIABLES
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0        SE,UE,VE,Us,Vs,Ss,Sb,Ts,Tb,AH,AV,Wx,Wy*
PAGE FORMATS
2 1                                           KPRNT1(DIGIT,CHAR), 2 (NEW PG)
CELLS WITH PRINT AT ALL LEVELS
2
035026 047022
CELLS IN LONGITUDINAL SECTION
1
ISLICE: JSLICE,MSNS
10 006017 019017 020018 023018 026021 028021 030023 031023 035027 038027
CELLS FOR LATER GRAPHING
13 4 0                                         IGPH,NSTGPH,IGPHOP      *
1 50 34 1           ocean boundary          L,M,N,ITYP
1 52 32 1           cbbt
1 32 18 1           Colonial Beach
1 11 13 1           baltimore
1 33 21 1           lewissetta
1 48 32 1           kiptopeake
1 48 24 1           Gloucester Pt
1 28 20 1           Solomons
1 15 16 1           Annapolis
1 52 32 7           x-stress # 1 (CBBT)
1 11 13 7           x-stress # 2 (Thomas Point)
1 52 32 8
1 11 13 8
----- TIME-VARIABLE BOUNDARY INPUTS-----
STARTING DATE
1994 1 1 00 00                           IYEAR,MONTH, IDAY, HOUR, MIN
=OCEAN WATER LEVELS
1
wl.dat
=WINDS
1
wind.dat.binary
=RIVER FLOWRATES
9
rv.dat
=OCEANIC SALINITIES
10
ocean_sal.dat
=OCEANIC TEMPERATURES
10
ocean_tmp.dat
=RIVER TEMPERATURES
1
river_tmp.dat
=ADDITIONAL MET DATA
4
meto.dat
=INITIAL CONDITIONS

```

```
0  
----- OUTPUT FILE NAMES-----*  
FPRINT  :m20.prn          *  
FGRAPH   :m20.gph          *  
FMED     :MECMED.DAT      *  
=====END OF FILE =====
```

APPENDIX D. CODE LISTING

Code listing for program MECCA2. Common blocks appear at the end.

```

1       PROGRAM MECCA2
2  CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
3  C
4  C Model for Estuarine and Coastal Circulation Assessment II
5  C
6  C PURPOSE - TO RUN A THREE-DIMENSIONAL (X-, Y-, Q-DIRECTION)
7  C HYDRO MODEL WITH NON-LINEAR ADVECTIVE TERMS AND
8  C VARIABLE DENSITY. USES NUMERICAL SCHEME OF ABBOTT
9  C FOR THE EXTERNAL (2-D) MODE, AND ANOTHER SCHEME FOR
10 C THE INTERNAL (BAROCLINIC) MODE. INCLUDES THOR
11 C AAUP's VARIABLE ARRAY FEATURES.
12 C
13 C LOGICAL UNITS USED
14 C
15 C     Unit Name   No.   Used for           Generic
16 C     -----      --   -----             -----
17 C INPUT:    LUCON      2    READ CONTROL FILE          FCON
18 C                 TO READ GEOGRAPHY FILE       FGE0
19 C                 READ INITIAL COND'S FILES  FINIT
20 C LUKB       5    KEYBOARD            -
21 C LUTID      21   TIDE BC              -
22 C LUWNND     22   GRIDDED WIND FILES      -
23 C LURIV      23   RIVER FLOWRATES        -
24 C LUSAL      24    OCEAN SALINITY BC       -
25 C LUOCT      25    OCEAN TEMPERATURE BC      -
26 C LURVT      26    RIVER TEMPERATURES        -
27 C LUMET      27   AIR PRESS, TMP, HUMIDITY      -
28 C
29 C OUTPUT:   ISCR      6    CRT SCREEN            -
30 C IO         10   LINE PRINTER          FPRINT
31 C LUGRF      11   DATA AT GRAPH POINTS      FGRAPH
32 C LUMED     12    INTERMEDIATE,FINAL VALUES   FMED
33 C
34 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
35 C
36 C FLOW CHART FOR MAIN PROGRAM
37 C
38 C MAIN
39 C :
40 C :-----YTIME8
41 C :-----NYEAR
42 C :
43 C :-----READZ-----ZEROS
44 C :      RDCON1
45 C :      RDCON2-----JULIAN
46 C :      RDCON3   IRR
47 C :      RDWIND
48 C :      RDGEO-----CHECKS
49 C :      RDICS
50 C :
51 C :-----INITS-----SETUP-----FLAGS
52 C :      SETSTP-----BSTATE-----RR
53 C :      THETAS
54 C :      WVERT
55 C :
56 C :-----OUTPUT-----PRNCON
57 C :      PRNBCG
58 C :      MEDSAV
59 C :      PGGRAPH
60 C :      PRINTA-----PRINTH-----DATES
61 C :      PRINTX-----PRNCHR
62 C <BEGIN LOOP>  PRINTI-----GRADP-----FRHO
63 C :      PSLICE
64 C :
65 C :-----FORCES-----ATMOS-----RDWIND
66 C :      BSTRES
67 C :      ALGRAD
68 C :
69 C :-----EXMODE-----THETAS
70 C :      BNDRY1-----RR
71 C :      HORVIS
72 C :      UHVV
73 C :
74 C :-----INTRNL-----BNDRY2
75 C :      VERVIS
76 C :      UPFP-----GRADP-----FRHO
77 C :      WVERT   GETCJ
78 C :
79 C :-----CONCZ-----BNDRY3-----BSTATE-----RR
80 C :      BNDRY4   RR
81 C :      HEAT1
82 C :      GFLUX-----HEATZ
83 C :
84 C :-----ANALYS
85 C :
86 C :-----OUTPUT-----PRNCON
87 C :      PRNBCG
88 C :      MEDSAV
89 C :      PGGRAPH
90 C :      PRINTA-----PRINTH-----DATES
91 C :      PRINTX-----PRNCHR
92 C <END LOOP>  PRINTI-----GRADP-----FRHO
93 C :      PSLICE
94 C (END)
95 C :
96 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
97 C
98 C VARIABLES
99 C
100 C     AG = GRAVITATIONAL ACCELERATION (M/S**2)
101 C     AH(,) = HORIZONTAL EDDY VISCOSITY (M**2/S) TIMES DEPTH
102 C             AT GRID CENTER
103 C     AHC(,) = HORIZONTAL EDDY VISCOSITY (M**2/S) TIMES DEPTH
104 C             AT GRID CORNER
105 C     AH3(,,) = 3-D HORIZONTAL EDDY VISCOSITY (M**2/S) TIMES DEPTH
106 C             AT GRID CENTER
107 C     AHMAX = MAXIMUM ALLOWABLE VALUE FOR AH(,)
108 C     AHO0 = INITIAL VALUE OF HORIZONTAL EDDY VISCOSITY
109 C     AREA(,) = FRACTION OF AREA OF GRID THAT IS WATER
110 C     AV(,,) = VERTICAL EDDY VISCOSITY (M**2/S)
111 C     AV00 = INITIAL VALUE OF VERTICAL EDDY VISCOSITY
112 C     BSNANG = ANGLE BASIN GRID MUST BE ROTATED CLOCKWISE TO BE

113 C
114 C     BSNLAT,BSNLON = NORTH LATITUDE, WEST LONGITUDE OF BASIN GRID POI
115 C     BX(,),BY(,) = WIDTH FACTOR IN X-, Y-DIRECTION
116 C     CAW1,CAW2 = AIR-WATER INTERFACIAL DRAG COEFFICIENTS
117 C     CRICH(1)-CRICH(8) = PARAMETERS IN RICHARDSON NUMBER REDUCTION OF
118 C                         DIFFUSION
119 C     CWB1,CWB2 = WATER-BOTTOM INTERFACIAL DRAG COEFFICIENTS
120 C     D(,) = METER SEA LEVEL DEPTH (M)
121 C     DENRAT = RATIO OF AIR DENSITY TO WATER DENSITY
122 C     DFDM,DFDN = CHANGE IN LOCAL CORIOLIS PARAMETER PER UNIT CHANGE
123 C                         IN GRID INDEX M, N TIMES .25*DTE)
124 C     DHAH = RATIO OF DH(M,M) TO AH(N,N)
125 C     DHMAX = MAXIMUM ALLOWABLE VALUE FOR DH(,)
126 C     DL = GRID LENGTH (M)
127 C     DMAX = MAX. DEPTH IN COMPUTATIONAL REGION
128 C     DPADX,DPADY = ATMOSPHERIC PRESSURE GRADIENT (MB/KM)
129 C     DQ = DIMENSIONLESS VERTICAL GRID INTERVAL
130 C     DTE = TIMESTEP (S) FOR EXTERNAL MODE
131 C     DTI = TIMESTEP (S) FOR INTERNAL MODE
132 C     DTMAX = MAX. TIMESTEP (S) FOR EXTERNAL MODE
133 C     DTIMAX = MAX. TIMESTEP (S) FOR INTERNAL MODE. UPDATED
134 C                         EACH TIME NEW TURBULENT VISCOSITY IS COMPUTED.
135 C     DV(,,) = VERTICAL MASS EXCHANGE COEFFICIENT
136 C     DV00 = INITIAL VALUE OF EDDY DIFFUSIVITY
137 C     E = SMALL AMOUNT ADDED TO PREVENT ZERO DIVIDE
138 C     FA(,),FB(,) = RECURSION ARRAYS USED FOR THE IMPLICIT SCHEME
139 C     FEDGE(,) = EDGE FUNCTION TO RAMP UP FORCES AT OCEAN BOUNDAR
140 C     GA(,),GB(,) = ARRAYS USED FOR THE IMPLICIT SCHEME
141 C     HCRCNC = INDEX FOR WHICH MUST ELAPSE BEFORE S, T
142 C                         ARE UPDATED
143 C     HR = HOURS PAIRED IN THIS SEGMENT
144 C     HRO = ELAPSED TIME (HRS) FROM PREVIOUS RUNS
145 C     HR1 = TOTAL ELAPSED TIME (HRS) = CUM HOURS
146 C     IBETAA = INDEX FOR INCLUSION OF NON-LINEAR ADVECTION
147 C     TERMS: (0=NO, 1=YES)
148 C     IBETAH = INDEX FOR NON-LINEAR FINITE-DEPTH TERMS (0=NO,
149 C                         1=YES)
150 C     IBETAP = INDEX TO INCLUDE HORIZONTAL PRESSURE GRADIENT
151 C                         DUE TO DENSITY STRUCTURE (0=NO, 1=YES)
152 C     IBOBV = INDEX FOR BOTTOM BC FOR THE INTERNAL-MODE VEL.
153 C     0 = U IS ZERO AT BOTTOM
154 C     1 = TAU=1ST-ORDER DU/DZ (DEFAULT WHEN INTER=0)
155 C     2 = TAU=2ND-ORDER DU/DZ
156 C     3 : AVDU/DZ = TAU log-layer, mid-level
157 C     ICOUPL = INDEX TO INCLUDE DENSITY VARIATIONS
158 C     0 = NO COUPLING
159 C     1 = VERTICAL VARIATIONS ONLY (AV,DV)
160 C     2 = VERT AND HORIZ (GRAD, GRX, GSTARK)
161 C     ICPOS = INDEX TO INSURE ZERO OR POSITIVE CONCENTRATIONS:
162 C     0 = NO, OTHERWISE = YES
163 C     ICOL(,),IROW(,) = ARRAYS OF COLUMN AND ROW FLAG NUMBERS
164 C     ICOR = INDEX FOR CORIOLIS ACCEL (0 OR LESS=NONE,
165 C                         1=CONSTANT, 2 OR MORE=BETA PLANE APPX.)
166 C     IXEXTN = INDEX FOR DOING THE EXTERNAL MODE CALCS (0=NO,
167 C                         OTHERWISE YES)
168 C     IFIELD(,) = INDEX FOR COMPUTATIONAL GRID=IJ
169 C     I = STATUS:0=LAND, 1=HALF FULL, 2=FULL, 3=WATER
170 C     LEVEL B.C., 4=RIVER FLOW B.C.
171 C     J = BARRIER INDEX:0=X & Y FLOW, 1=NO X FLOW,
172 C     2=NO Y FLOW, 3=NO X, NO Y FLOW.
173 C     IGHPH = NUMBER OF QUANTITIES TO BE GRAPHS
174 C     IHATE = INDEX FOR ATMOSPHERIC HEATING:
175 C     1=NO HEAT FLUX
176 C     2=NORMAL HEATING
177 C     IHIVISC = INTERVAL FOR UPDATING HORIZONTAL TURBULENT
178 C     VISCOSITY (DTA-TIMESTEPS)
179 C     ILFT = INDEX FOR LEFT-MOST BOUNDARY CONDITION:
180 C     0 = WATER LEVEL (TIDE OR RADIATION CONDITION)
181 C     1 = FLOWRATE (RIVER)
182 C     2 = ZERO FLOWRATE (SOLID WALL)
183 C     INTER = INDEX FOR COMPUTING INTERNAL MODE VARIABLES
184 C     (0=NO, OTHERWISE = YES)
185 C     IO = PRINT OUTPUT CHANNEL NUMBER
186 C     IRGT = INDEX FOR RIGHT-MOST BOUNDARY CONDITION:
187 C     0 = WATER LEVEL (TIDE OR RADIATION CONDITION)
188 C     1 = FLOWRATE (RIVER)
189 C     2 = ZERO FLOWRATE (SOLID WALL)
190 C     ISCR = CRT SCREEN OUTPUT CHANNEL NUMBER
191 C     ISPLIT = NO. OF EXTERNAL TIMESTEPS PER INTERNAL Timestep
192 C     ITOPV = INDEX FOR TOP B.C. FOR INTERNAL MODE VELOCITY
193 C     ITPO(1) = DIRECTION OF CUTFLOW: 1=X, -1=-X, 2=Y, -2=-Y
194 C     ITPO() = TYPE OF OCEAN BOUNDARY CONDITION
195 C     1 : WATER LEVEL SPECIFICATION
196 C     2 : RADIATION OUTFLOW
197 C     3 : RADIATION CONDITION, ORLANSKI
198 C     4 : RADIATION CONDITION, RIEMANN
199 C     ITPR() = RIVER DIRECTION TYPE: 1=FLOW IN X, 2=FLOW IN Y
200 C     (+=FLOW IN POSITIVE DIRECTION, -=FLOW IN NEG. DIR)
201 C     ITYP() = TYPE OF QUANTITY TO BE GRAPHED AT LGPH, MGPH,
202 C     NGPH:
203 C     1 : WATER LEVEL
204 C     2 : UE
205 C     3 : VE
206 C     4 : UE + U
207 C     5 : VE + V
208 C     6 : W
209 C     7 : WIND STRESS IN X DIRECTION
210 C     8 : WIND STRESS IN Y DIRECTION
211 C     9 : SALINITY
212 C     10: TEMPERATURE (CELSIUS)
213 C     11: AH3/H
214 C     12: THETA1
215 C     13: THETA2
216 C     14: THETA3
217 C     15: AV
218 C     16: DV
219 C     17: RI
220 C     18: WIND IN X DIRECTION
221 C     19: WIND IN Y DIRECTION
222 C     20: UHOLD/H

```



```

477 DO 110 N=NBL(I),NB2(I)
478 JFIELD(N,M)=30+MOD(JFIELD(N,M),10)
479 IF(JTP0(I).EQ.2)JFIELD(N,M)=40+MOD(JFIELD(N,M),10)
480 110 CONTINUE
481 C      RIVERS
482 120 IF(NUMRIV.LE.0)GOTO 140
483 DO 130 I=1,NUMRTV
484 DO 130 M=M1(I),MR2(I)
485 DO 130 N=N1(I),NR2(I)
486 IF(JTPR(I).EQ.1)JFIELD(N,M)=40+MOD(JFIELD(N,M),10)
487 IF(JTPR(I).EQ.2)JFIELD(N,M)=20+MOD(JFIELD(N,M),10)
488 130 CONTINUE
489
490 C      FIND THE ICOL PARAMETERS
491 140 DO 150 J=1,NM2SIZ
492 DO 150 I=1,5
493 ICOL(I,J)=0
494 150 IROW(I,J)=0
495 ICOUNT=0
496 NCOL=0
497 DO 190 N=1,NMAX
498 ISTART=0
499 DO 190 M=1,MMAX
500 IC=JFIELD(N,M)/10
501 IF(IC.EQ.0.OR.IC.GE.3)GOTO 190
502 C      INSERT GRID LINE SPECS IF THIS GRID STARTS A LINE
503 IF(ISTART.EQ.1)GOTO 170
504 ISTART=1
505 NCOL=NCOL+1
506 C      DO A FIX-UP IF NCOL IS TOO LARGE FOR ARRAY
507 IF(NCOL.LE.NM2SIZ)GOTO 160
508 NCOL=NM2SIZ
509 ICOUNT=ICOUNT+1
510 160 ICOL(1,NCOL)=N
511 ICOL(2,NCOL)=M
512 C      DETERMINE LOWER END BOUNDARY CONDITION
513 ICOL(4,NCOL)=2
514 IF(M.EQ.1)GOTO 170
515 I=JFIELD(N,M-1)/10
516 J=JFIELD(N,M-1)-10*I
517 IF(J.EQ.1.OR.J.EQ.3)GOTO 170
518 IF(I.EQ.3)ICOL(4,NCOL)=0
519 IF(I.EQ.4)ICOL(4,NCOL)=1
520 IF(I.EQ.5)ICOL(4,NCOL)=2
521 C      CHECK FOR UPPER END BOUNDARY CONDITIONS
522 170 ICOL(3,NCOL)=M
523 ICOL(5,NCOL)=2
524 IF(M.EQ.NMAX)GOTO 180
525 J=JFIELD(N,M)-10*(JFIELD(N,M)/10)
526 IF(J.EQ.1.OR.J.EQ.3)GOTO 180
527 IP=JFIELD(N,M-1)/10
528 IF(IP.GT.0.AND.IP.LT.3)GOTO 190
529 IF(IP.EQ.3)ICOL(5,NCOL)=0
530 IF(IP.EQ.4)ICOL(5,NCOL)=1
531 IF(IP.EQ.5)ICOL(5,NCOL)=2
532 180 ISTART=0
533 190 CONTINUE
534 C      FIND THE IROW PARAMETERS
535 200 JCOUNT=0
536 NROW=0
537 DO 240 M=1,MMAX
538 ISTART=0
539 DO 240 N=1,NMAX
540 IC=JFIELD(N,M)/10
541 IF(IC.EQ.0.OR.IC.GE.3)GOTO 240
542 C      BEGIN TO INSERT LINE SPECS
543 IF(ISTART.GT.0)GOTO 220
544 ISTART=1
545 NROW=NROW+1
546 C      DO A FIX-UP IF NECESSARY
547 IF(NROW.LE.NM2SIZ)GOTO 210
548 NROW=NM2SIZ
549 JCOUNT=JCOUNT+1
550 210 IROW(1,NROW)=M
551 IROW(2,NROW)=N
552 C      DETERMINE LOWER END BOUNDARY CONDITION
553 IROW(4,NROW)=2
554 IF(N.EQ.1)GOTO 220
555 I=JFIELD(N-1,M)/10
556 J=JFIELD(N-1,M)-10*I
557 IF(J.EQ.2.OR.J.EQ.3)GOTO 220
558 IF(I.EQ.3)IROW(4,NROW)=0
559 IF(I.GE.4)IROW(4,NROW)=1
560 IF(I.GE.5)IROW(4,NROW)=2
561 C      CHECK FOR UPPER END BOUNDARY
562 220 IROW(3,NROW)=N
563 IROW(5,NROW)=2
564 IF(N.EQ.NMAX)GOTO 230
565 J=JFIELD(N,M)-10*(JFIELD(N,M)/10)
566 IF(J.EQ.1.OR.J.EQ.3)GOTO 230
567 IP=JFIELD(N+1,M)/10
568 IF(IP.GT.0.AND.IP.LT.3)GOTO 240
569 IF(IP.GE.3)IROW(5,NROW)=0
570 IF(IP.GE.4)IROW(5,NROW)=1
571 IF(IP.GE.5)IROW(5,NROW)=2
572 230 ISTART=0
573 240 CONTINUE
574 C      CHECK THE COUNTS
575 280 IF(ICOUNT.GT.0)WRITE(IO,290)NM2SIZ,ICOUNT
576 290 FORMAT(5X,'*** ERROR IN FLAGS: NUMBER OF ROW STRINGS EXCEEDS ',I4,
577 1 ' BY ',I4)
578 IF(JCOUNT.GT.0)WRITE(IO,295)NM2SIZ,JCOUNT
579 295 FORMAT(5X,'*** ERROR IN FLAGS: NUMBER OF COLUMN STRINGS EXCEEDS ',
580 1 I4,' BY ',I4)
581 C      GET FIRST, LAST COLUMN IN EACH ROW.
582 DO 310 M=1,MMAX
583 NA=NMAX+1
584 NB=1
585 DO 300 N=1,NMAX
586 IF(IFIELD(N,M)/10.EQ.0)GOTO 300
587 NA=MIN0(NA,NA)
588 NB=MAX0(NA,NB)
589 300 CONTINUE
590 NAB(M)=1000*NA+NB
591 310 CONTINUE
592 C      CREATE NEW IFIELD BY INSERTING RIGHT SIDE AND
593 C      BOTTOM SIDE BARRIERS IN WATER CELLS
594 DO 320 M=1,MMAX
595 DO 320 N=1,NMAX
596 IF(IFIELD(N,M)/10.EQ.0)GOTO 320
597 IF(IFIELD(N,M)/10.EQ.KOCNBT)GOTO 320
598 C      IF(IFIELD(N,M)/10.EQ.KRIVBC)GOTO 320
600 IX=0
601 IF(M.EQ.MMAX.OR.IFIELD(N,MIN0(M+1,MMAX))/10.EQ.0)IX=1
602 IF(MOD(IFIELD(N,M),10).EQ.1.OR.MOD(IFIELD(N,M),10).EQ.3)IX=1
603 IY=0
604 IF(N.EQ.NMAX.OR.IFIELD(MIN0(N+1,NMAX),M)/10.EQ.0)IY=2
605 IF(MOD(IFIELD(N,M),10).EQ.2.OR.MOD(IFIELD(N,M),10).EQ.3)IY=2
606 IFIELD(N,M)=10*(IFIELD(N,M)/10)+IX+IY
607 320 CONTINUE
608 C      OCEAN BOUNDARY CELLS
609 IF(NUMOBC.LE.0)GOTO 400
610 DO 390 I=1,NUMOBC
611 IX=0
612 IY=0
613 IF(IABS(ITPO(I)).EQ.1)THEN
614 IY=2
615 IF(ITPO(I).GT.0.AND.IFIELD(N,M+1).LT.9)IX=1
616 ENDIF
617 IF(IABS(ITPO(I)).EQ.2)THEN
618 IX=1
619 IF(ITPO(I).GT.0.AND.IFIELD(N+1,M).LT.9)IY=2
620 ENDIF
621 DO 390 M=M1(I),MB2(I)
622 DO 390 N=N1(I),NR2(I)
623 390 IFIELD(N,M)=10*KOCNBT+IX+IY
624 C      RIVERS
625 400 IF(NUMRIV.LE.0)GOTO 430
626 DO 420 I=1,NUMRIV
627 IF(JTPR(I).NE.2)GOTO 420
628 IX=0
629 IY=0
630 IF(IABS(ITPR(I)).EQ.1)IY=2
631 IF(IABS(ITPR(I)).EQ.2)IX=1
632 DO 410 M=M1(I),MR2(I)
633 DO 410 N=N1(I),NR2(I)
634 410 IFIELD(N,M)=10*KRIVBC+IX+IY
635 420 CONTINUE
636 430 CONTINUE
637 C      MFUX(,) = INDEX FOR MASS FLUX IN X-DIRECTION (0=NO, 1=YES)
638 C      NFUX(,) = INDEX FOR MASS FLUX IN Y-DIRECTION (0=NO, 1=YES)
639 C      LOOP DOWN THE LINES
640 DO 450 M=1,MMAX
641 MP=MIN0(MMAX,M+1)
642 DO 450 N=1,NMAX
643 NP=MIN0(NMAX,N+1)
644 I=MOD(IFIELD(N,M),10)
645 IX=0
646 IF(IFIELD(N,M).LT.10.OR.IFIELD(N,MP).LT.10)IX=0
647 IF(I.EQ.1.OR.I.EQ.3.OR.M.EQ.MMAX)IX=0
648 MFUX(N,M)=IX
649 IY=1
650 IF(IFIELD(N,M).LT.10.OR.IFIELD(N,MP).LT.10)IY=0
651 IF(I.EQ.2.OR.I.EQ.3.OR.N.EQ.NMAX)IY=0
652 NFLUX(N,M)=IY
653 450 CONTINUE
654 RETURN
655 END
656 C -----
657 C-----SUBROUTINE SETUP
658 C-----SUBROUTINE SETUP
659
660
661 C      MAY 1984 (REVISED APRIL 88) K. W. HESS
662 C      PURPOSE - TO INITIALIZE THE PARAMETERS AFTER ALL INPUT
663 C      FILES HAVE BEEN READ IN.
664 C      VARIABLES -
665 C      DPADX = GRADIENT OF ATMOS. PRESSURE IN X-DIRECTION
666 C      (NUITS = MB/KM = (100 N/M^2)/(1000 M)
667 C      SO DPADX*(./RHOU) HAS MKS UNITS
668 C      ICS = INDEX FOR READ IN OF INITIAL CONDITIONS
669 C      (0=NO, 1=YES)
670
671 INCLUDE 'COMM20.FOR'
672 C      CHECK BOTTOM DRAG
673 IF(INTER.EQ.0)IBOTW=1
674 C      SET CORIOLIS PARAMETERS
675 COR=0.0
676 IF(ICOR.NE.0)COR=2.*OMEGA*SIN(RAD*BSNLAT)
677 FCOR0=DCOR*COR
678 DFDM=0.0
679 DFDN=0.0
680 IF(IABS(ICOR).GE.2)THEN
681 CC=-DTB*2.*OMEGA*COS(BSNLAT*RAD)*DL*RAD/60./1852.
682 DFDM=CC*COS(BSNANG*RAD)
683 DFDN=CC*SIN(BSNANG*RAD)
684 END IF
685 C      SET DENSITY COUPLING
686 ICOUPL=MIN0(MAX0(0,ICOPL),2)
687 C      COMPUTE VERTICAL GRID PARAMETERS
688 DO=1./FLOAT(LAYRS)
689 HALFDQ=.5*DO
690 TWODQ=2.*DO
691 LBOT=LAYER+1
692 C      GET IFIELD
693 CALL FLAGS ! newest version
694 C      INITIALIZE VISCOSITIES AND GET MAX, MIN CELL DEPTHS
695 DMAZ=0.0
696 DMIN=1.E-10
697 DO 130 M=1,MMAX
698 DO 130 N=1,NMAX
699 IF(IFIELD(N,M).LT.10)GOTO 130
700 DMAZ=AMAX1(DMAX,D(N,M))
701 DMIN=AMIN1(DMIN,D(N,M))
702 IF(ICS.EQ.1)GOTO 130
703 AH(N,M)=D(N,M)*AH00
704 AH(C,N,M)=D(N,M)*AH00
705 DO 120 L=1,LBOT
706 AH3(L,N,M)=AH(N,M)
707 S(L,N,M)=SAL0
708 T(L,N,M)=TMO
709 IF(L.LT.LBOT)DV(L,N,M)=DV00
710 120 IF(L.LT.LBOT)AV(L,N,M)=AV00
711 130 CONTINUE
712
713 C      GET VERTICAL INTEGRATION COEFFICIENT FOR LEVELS (NOT LAYRS)
714 148 DO 160 L=1,LBOT
715 CT(L)=1./FLOAT(LBOT-1)
716 IF(L.EQ.1.OR.L.EQ.LBOT)CI(L)=0.5/FLOAT(LBOT-1)
717 160 CONTINUE
718 C      EDGE FUNCTION
719 DO 250 M=1,MMAX
720 DO 250 N=1,NMAX
721 FEDGE(N,M)=1.0
722 II=IFIELD(N,M)/10
723 IF(II.EQ.KOCNBT)FEDGE(N,M)=0.0
724 IF(M.EQ.1.OR.M.EQ.MMAX)GOTO 250
725 IF(N.EQ.1.OR.N.EQ.NMAX)GOTO 250
726 IF(II.EQ.KRIVBC)GOTO 250
727 IF(IFIELD(N,M)/10.EQ.KOCNBT.COR.IFIELD(N,M-1)/10.EQ.KOCNBT.
728 1 OR.IFIELD(N+1,M)/10.EQ.KOCNBT.COR.IFIELD(N-1,M)/10.EQ.KOCNBT)
```

```

729      2 FEDGE(N,M)=0.5
730   250 CONTINUE
731   RETURN
732   END
733 C-----+
734 C MECCA FILE : MPRINT
735 C-----+
736 C
737   SUBROUTINE OUTPUT
738 C     APRIL 1988 HESS MEAD VAX
739 C     PURPOSE - TO PRINT OUT VARIABLES AT START, OTHER SELECTED TIMES
740 C           AND END
741   INCLUDE 'COMM20.FOR'
742 C
743 C     INITIAL PRINTOUTS
744   IF(NSTI.EQ.0)CALL PRNCON(1)
745   IF(NSTI.EQ.0)CALL PRNBGG(1)
746 C     SAVE INTERMEDIATE RESULTS EACH HRSAVE HOURS
747   IF(NSTI.GT.0.AND.MOD(NSTI,IHR*IFIX(HRSAVE)).EQ.0)CALL MEDSAV
748 C     STORE VALUES FOR GRAPHING
749   CALL PGRAFH
750 C     PRINT OUT SELECTED FIELDS
751   CALL PGRAFH
752 C     END OF RUN
753   IF(ISTOP.EQ.1)GOTO 120
754   IF(NSTI.LT.NSTIMX)GOTO 700
755 C     WRITE HOURS TO CONSOLE
756   120 NRR=NN+1IFIX(HB0)
757   CALL PBNCN(0)
758   WRITE(IO,525)
759   525 FORMAT(//,1X,'VII. RUN COMPLETION',55(''),/)
760   WRITE(IO,520)NEGS
761   520 FORMAT(5X,'NO. OF TIMES A WATER TOTAL DEPTH WENT NEGATIVE=',I4)
762   WRITE(IO,530)FPRINT,FMED
763   WRITE(ISCR,530)EPRINT,FMED
764   530 FORMAT(5X,'PRINT OUTPUT IS IN FILE : ',A40,
765   1      5X,'VELOCITY DATA IS IN FILE : ',A40)
766   IF(IGPH.GT.1)WRITE(ISCR,540)FGRAPH
767   IF(IGPH.GT.1)WRITE(IO,540)FGRAPH
768   540 FORMAT(5X,'GRAPHING OUTPUT IS IN : ',A40)
769   630 FORMAT(1X,'RUN COMPLETE')
770   110 IO,NE,ISCR,NAME(IO,640)
771   640 FORMAT(1X,'RUN COMPLETE',//,1X,74(''))
772   WRITE(IO,'(*')ISTOP',ISTOP
773   WRITE(ISCR,'(*')ISTOP',ISTOP
774   700 CONTINUE
775   RETURN
776   END
777 C-----+
778 C
779 C-----+
780 C     SUBROUTINE MEDSAV
781 C     PURPOSE - TO SAVE RESULTS TO AN INTERMEDIATE FILE
782 C
783 C     INCLUDE 'COMM20.FOR'
784 C     NOTIFY PRINT FILE
785 C     WRITE(IO,110)NSTI,UT
786   110 FORMAT(//,1X,'MEDSAV: NSTI=',I5,' UT=',F10.4)
787   110 OPEN FILES
788   CALL FUOPEN(LUMED,FMED)
789   C
790   C     WRITE TO DATA FILE
791   K=1+KONGEN
792   WRITE(LUMED)NMAX,MMAX,LBOT,NSTET,UT,YEAR,K
793   WRITE(LUMED)SE,UE,VE,SOLD,UHOLD,VHOLD,AH,AV,PHI,TBX,TBY,
794   1 U,V,W,THETA1,THETA2,THETA3
795   WRITE(LUMED,63)WX,WI,TSX,TSY
796   IF(KONGEN.GT.0)WRITE(LUMED)S,T,DV,RI,NSTINF
797 C     CLOSE FILES
798   CLOSE(LUMED)
799   120 RETURN
800   END
801 C-----+
802 C
803 C-----+
804   SUBROUTINE PRINTA
805 C     1986 MEAD K.HESS VAX
806 C     PURPOSE - TO CALL THE VARIOUS OCCASIONAL PRINTING SUBROUTINES
807 C     VARIABLES -
808 C       HROUTO = HOUR OF FIRST P/O
809   INCLUDE 'COMM20.FOR'
810   NOUTO=AMAX1(1.,HROUTO*3600.01/DTI)
811   NOUT1=AMAX1(1.,HROUTO*3600.01/DTI)
812   IF(HR1.LT.HROUTO)GOTO 120
813 C     IF(.NOT.(NSTI.EQ.NSTIMX).OR.MOD(NSTIT-NOUT0,NOUT1).EQ.0.OR.
814   IF(.NOT.(NSTI.EQ.NSTIMX).OR.MOD(NSTIT,NOUT1).EQ.0.OR.
815   1 NSTI.EQ.0.OR.ISTOP.EQ.1))GOTO 110
816   write(6,*)'PRINTA: NOUT0,NOUT1=',nout0,nout1
817   IF(NSTI.EQ.0)WRITE(IO,100)
818   100 FORMAT(//,1X,'IV. INITIAL FIELDS',55(''))
819   C     PRINT (8) TOP VIEW OF EXTERNAL MODE
820   CALL PRINTX
821   C     PRINT (C) ALONG A SLICE
822   CALL PSLICE
823   C     PRINT (D) AT ALL LEVELS AT ONE CELL
824   CALL PRINTI
825   110 CONTINUE
826   120 CONTINUE
827   IF(NSTI.EQ.0)WRITE(IO,130)
828   130 FORMAT(//,1X,'V. RUN-TIME OUTPUT',55(''))
829   RETURN
830   END
831 C-----+
832 C
833 C-----+
834   SUBROUTINE PRINTI
835 C     MAY 1984 K. W. HESS
836 C     PURPOSE - TO PRINT OUT THE INTERNAL-MODE VARIABLES.
837   INCLUDE 'COMM20.FOR'
838   DIMENSION RJ(7),FFX(LSIZE),FPY(LSIZE),WT(LSIZE),TRAD(LSIZE)
839 C     CHECK FOR PRINT INDEX
840   IF(IPRMN.LE.0)GOTO 450
841   WRITE(IO,100)UWT
842   100 FORMAT(//,1X,'D. INTERNAL MODE VARIABLES AT ALL LEVELS IN A CELL',
843   1 ' AT UT=',F9.4)
844   WFACT=10.*NINT(ALOG10(DL/DMAX/DQ))
845 C     GET ROW OR COLUMN INDICES
846   DO 440 I=1,NPRMN
847   M=IPRMN(I)/1000
848   N=IPRMN(I)-1000*M
849   MP=MIN0(N+1,MMAX)
850   NP=MIN0(N+1,NMAX)
851 C     CHECK FOR CELL INSIDE WATER FIELD
852   IF(N.GE.1.AND.M.GE.1.AND.N.LE.NMAX.AND.M.LE.MMAX)GOTO 120
853   WRITE(IO,110)N,M
854   110 FORMAT(3X,'ERROR: CELL N='',I2,',M='',I2,' IS OUTSIDE GRIDMESH')
855   GOTO 440
856   120 IF(IFIELD(N,M).GE.10)GOTO 140
857   WRJTB(IO,130)N,M
858   130 FORMAT(3X,'ERROR: CELL N='',I3,',M='',I3,' IS NOT WATER')
859   GOTO 440
860   140 CONTINUE
861   WRITE(IO,150)N,M,UT,NSTI,WFACT
862   150 FORMAT(//,1X,'N='',I3,',M='',I3,' UT='',F10.4,'NSTI='',I6,
863   1 'L,'f',6x,'UTm',7X,'UTp',7X,'VTm',7X,'VTp',7X,'Wxf',7X,
864   2 'S ',7x,' T ')
865   IF(M.LE.0.OR.N.LE.0)GOTO 440
866   C     VELOCITIES
867   DHDT1=(SE(N,M)-SOLD(N,M))/DTI
868   DO 160 L=1,LBOT
869   R(J(1))=UE(N,M-1)+U(L,N,M-1)
870   R(J(2))=UE(N,M)+U(L,N,M)
871   R(J(3))=VE(N-1,M)+V(L,N-1,M)
872   R(J(4))=VE(N,M)+V(L,N,M)
873   Q=FLOAT(1-L)*DQ
874   WT(L)=W(L,N,M)+(1.+Q)*DHDT1
875   R(J(5))=W(L,N,M)*WFAC
876   R(J(6))=L,N,M)*WFAC
877   R(J(7))=R(L,N,M)
878   R(J(8))=R(L,N,M)
879   WRITE(IO,155)L,(R(J(K)),K=1,7)
880   155 FORMAT(2X,I2,7F10.3,F10.4)
881   160 CONTINUE
882 C     INTERNAL HEATING
883   FS=0.0
884   FO=0.0
885   IF(NSTI.GT.1)ISPLIT.AND.IHEAT.GT.0.AND.KONCEN.GE.2
886   1 CALL HEAT(N,M,TRAD,FTSURF)
887   WRITE(IO,170)
888   170 FORMAT(1X,' L',4X,'10+4 Av ',2X,'10+4 Dv ',5X,'Ri ',7X,
889   1 'Trad ',6X,'AH3 ',6X,'Rho ',5x,'10+6 Tau ')
890 C     DIFFUSIVITY, DENSITY, STRESS
891   D2=2.*DQ*(D(N,M)+SE(N,M))
892   DO 195 L=1,LAYRS
893   R(J(1))=AV(L,N,M)*10000.
894   R(J(2))=DV(L,N,M)*10000.
895   R(J(3))=RI(L,N,M)
896   TRAD(L)=D(L)
897   TW=L,1,N,M
898   R(J(4))=TRAD(L)
899   R(J(5))=AH3(N,M)/(D(N,M)+SE(N,M))
900   R(J(6))=UHOLD(N,M)*FRHO(G(S(N,M),T(L,N,M)))
901   R(J(7))=1.E+6*AV(L,N,M)*(U(L,N,M)-U(L+1,N,M)+U(L,N,M-1)
902   1 -U(L+1,N,M-1)+V(L,N,M)-V(L+1,N,M)+V(L,N-1,M)-V(L+1,N-1,M))/DZ
903   WRITE(IO,180)L,(R(J(K)),K=1,7)
904   c180 FORMAT(2X,I2,5F10.3,3X,F10.4)
905   180 FORMAT(2X,I2,7F10.3)
906   195 CONTINUE
907 C     TWO-DIMENSIONAL VARIABLES
908 C     MP=MIN0(M+1,MMAX)
909   NP=MIN0(N+1,NMAX)
910   H1=0.5*(D(N,M)+SE(N,M)+D(N,MP)+SE(N,MP)+E)
911   H2=0.5*(D(N,M)+SE(N,M)+D(N,MP)+SE(N,MP)+E)
912   C     PRINT OTHER VARIABLES, LIKE DEPTH AND BOTTOM STRESSES
913 C     FAH=AH(N,M)/(D(N,M)+SE(N,M))
914   UO=UHOLD(N,M)/H2
915   VO=VHOLD(N,M)/H2
916   UBOT=U(LBOT,N,M)
917   VBOT=V(LBOT,N,M)
918   LMID=MAX0(WBOT,UBOT)/2
919   PHIX=.5*(PHI1(N,M)+PHI(N,M+1))
920   PHIN=.5*(PHI1(N,M)+PHI(N+1,M))
921   WRITE(IO,250)SE(N,M),UE(N,M),VE(N,M),D(N,M),TBX(N,M),TBY(N,M),
922   1 Q1,QA,QB,QS,QE,
923   2 GSTARX(N,M),GSTARY(N,M),AREA(N,M),BX(N,M),BY(N,M),
924   3 PHI(N,M),VO,VO,SOLD(N,M),FAH,
925   4 THETA1(N,M),THETA2(N,M),THETA3(N,M),
926   5 THETSU(N,M),THETSV(N,M),FEDGE(N,M),
927   6 UBOT,VBOT,LMID,W(LMID,N,M),TSX(N,M),TSY(N,M),
928   7 DPADX,DPADY,SE(N,MP),SE(NP,MP),AHC(N,M),PHIX,PHIY
929   250 FORMAT(3X,' SE='',F8.5,' UE='',F8.5,
930   1 ' D='',F7.2,' TBX='',2E11.4,'/,3X,' QI='',E9.3,
931   2 ' QA='',E9.3,' QB='',E3.3,' QS='',E9.3,' QE='',E9.3,
932   3 '/,3X,' GSTARX,Y='',2E11.4,' AREA='',F5.3,
933   4 ' BX='',F5.3,' BY='',F5.3,'/,3X,' PHI='',F7.5,
934   5 ' UHOLD/D='',F7.3,' VHOLD/D='',F7.3,' SOLD='',F6.3,' AH/D='',F9.3,
935   6 '/,3X,' THETA1,2,3,'/,3F7.3,' THETSU,V='',2E10.3,' EDG='',F3.1,'/
936   7 ' 3X,' U,VBOT='',2E11.4,' W(I2,'),'',E10.4,' TSX,Y='',2E9.2,
937   8 '/,3X,' DPADX,Y='',2E9.2,' SEMP,SENp='',F7.4,F8.4,' AHC='',F9.3,
938   9 '/,3X,' PHIX,Y='',F210.7)
939   440 CONTINUE
940   440 RETURN
941   450 END
942 C-----+
943 C
944 C-----+
945 C
946 C-----+
947 C     SUBROUTINE PRINTX
948 C     MAY 1984 K. W. HESS MEAD VAX11/750
949 C     PURPOSE - TO PRINT OUT THE TOP VIEWS OF EXTERNAL VARIABLES
950 C     VARIABLES - I = VARIABLE INDEX
951 C       1 = WATER LEVEL
952 C       2 = U MEAN VELOCITY
953 C       3 = V MEAN VELOCITY
954 C       4 = TOTAL SURFACE X VELOCITY
955 C       5 = TOTAL SURFACE Y VELOCITY
956 C       6 = SURFACE SALINITY
957 C       7 = BOTTOM SALINITY
958 C       8 = SURFACE TEMPERATURES (C)
959 C       9 = BOTTOM TEMPERATURES (C)
960 C      10 = HORIZONTAL VISCOSITY
961 C      11 = VERTICAL VISCOSITY
962 C      12 = WX
963 C      13 = WY
964 C
965 C     JPRINT=1 PRINT INDEX: IF=1 , PRINT
966 C     INCLUDE 'COMM20.FOR'
967 C     DIMENSION MN(NSIZE),NO(NSIZE),JFIELD(NSIZE)
968 C     CHARACTER*1 ANUM(3*NSIZE)
969 C     CHARACTER*43 QTITLE(I3)
970 C     DATA QTITLE/
971 C       1 'WATER LEVELS (CM) ',
972 C       2 'X-DIR VERTICALLY-AVERAGED VELOCITIES (CM/S) ',
973 C       3 'Y-DIR VERTICALLY-AVERAGED VELOCITIES (CM/S) ',
974 C       3 'X-DIR TOTAL SURFACE VELOCITIES (CM/S) ',
975 C       3 'Y-DIR TOTAL SURFACE VELOCITIES (CM/S) ',
976 C       3 'SURFACE SALINITIES (PPT) ',
977 C       5 'SURFACE TEMPERATURES (DEG. C) ',
978 C       5 'BOTTOM TEMPERATURES (DEG. C) ',
979 C       5 'HORIZONTAL VISCOSITY (M**2/S) ',
980 C       5 'VERTICAL VISCOSITY x 100 (M**2/S) '

```

```

981      5 'WX (M/S)
982      5 'WY (M/S)
983      DIMENSION LEV(13),ISEL(13),FAC(13)
984      DATA LEV/1.,1.,1.,1.,1.,1.,1.,1./
985      DATA FAC/100.,100.,100.,100.,1.,1.,1.,1.,1.,1.,1.,1./
986      DATA ISEL/1,2,3,4,5,9,9,10,10,11,15,18,19/
987      LEV(7)=LBOT-1
988      LEV(9)=LBOT-1
989      LEV(11)=LBOT/2
990      NMAX=3*NMAX
991      DO 100 N=1,NMAX
992      100 NO(N)=N
993      C      LOOP THRU THE VARIABLES
994      ICALL=0
995      DO 250 I=1,13
996      IF(JPRNT(I).NE.1)GOTO 250
997      C      PRINT PAGE FORM
998      IF(KPRNT2.NE.2)WRITE(10,110)
999      110 FORMAT(1X)
1000     IF(KPRNT2.EQ.2)WRITE(10,120)
1001     120 FORMAT(1H1)
1002     C      WRITE DESCRIPTOR LINE
1003     ICALL=ICALL+1
1004     IF(ICALL.EQ.1)WRITE(10,125)
1005     125 FORMAT(//1X,'B. TOP VIEW OF SELECTED VARIABLE FIELDS')
1006     C      WRITE(10,130)QTITLE(I),UT,NSTI,HR,HRL,CUMDAY
1007     C130 FORMAT(10,A43,/,1X,'UT=',F10.4,'NSTI=',I5,'HR=',F8.2,'CUM.HR='
1008     RELDAY=HH/24.
1009     C      WRITE(10,130)QTITLE(I),YEAR,UT,RELDAY,CUMDAY,NSTI
1010     C130 FORMAT(2X,A43,/,1X,'YEAR',F5.0,'UT',F10.4,
1011     1,'REL.DAY',F8.2,'CUM.DAY',F8.2,'NSTI',I5)
1012     NPL=25          ! number per line
1013     KMAX=1+(NMAX-1)/NPL
1014     DO 220 K=1,KMAX
1015     N1=N+1+NPL*(K-1)
1016     N2=MINO(NMAX,N1+NPL-1)
1017     C      WRITE(10,134)(NO(N),N=N1,N2)
1018     134 FORMAT(4X,I0,3,(5X,40I3))
1019     C      LOOP THRU ARRAY AND GET VALUE
1020     DO 180 M=1,MMAX
1021     DO 136 N=1,NMAX
1022     136 NUM(N)=0
1023     DO 140 N=NL,N2
1024     JFIELD=N-JFIELD(N,M)
1025     140 NUM(N)=NINT(FAC(I)*SELECT(ISEL(I),LEV(I),N,M))
1026     C      OPTION 1: PRINT OUT VECTOR AS STRAIGHT NUMBERS
1027     IF(KPRNT1.EQ.2)GOTO 160
1028     C      WRITE(10,150)M,(NUM(N),N=N1,N2)
1029     150 FORMAT(1X,I3,40I5,3/(4X,40I5))
1030     GOTO 180
1031     C      OPTION 2: PRINT OUT VECTORS AS CHARACTERS, WITH ." FOR LAND
1032     CALL PRNCHR(3,NUM,JFIELD,ANUM,NMAX)
1033     N1A=1+3*(NL-1)
1034     N2A=MINO(3*NMAX,N1A-1+3*NPL)
1035     C      WRITE(10,170)M,(ANUM(N),N=N1A,N2A)
1036     170 FORMAT(1X,I3,140A1,3/(5X,140A1))
1037     180 CONTINUE
1038     220 CONTINUE
1039     250 CONTINUE
1040     RETURN
1041     END
1042 C-----+
1043 C-----+
1044 C-----+
1045      SUBROUTINE PRNCHR(MAXDIG,NUM,JFIELD,ANUM,KMAX)
1046      C      OCTOBER 1984 K. W. HESS MEAD VAX11/750
1047      C      PURPOSE - TO CONVERT NUMERIC VALUES TO CHARACTER VALUES.
1048      C      IT'S ASSUMED THAT THE OUTPUT NUMBER IS AT MOST
1049      C      3 *(MAXDIG) DIGITS LONG.
1050      INCLUDE 'COMM20.FOR'
1051      DIMENSION NUM(NPMSIZ),IMAX(4),JFIELD(NPMSIZ)
1052      CHARACTER*1 ANUM(3*NPMSIZ)
1053      CHARACTER*1 DIGIT(10),CHAR(5)
1054      DATA DIGIT/'0','1','2','3','4','5','6','7','8','9'/
1055      DATA CHAR/' ',' ',',','*','/','.'/
1056      DATA IMAX/9,99,999,9999/
1057      C      LOOP THRU THE NUMBERS
1058      DO 200 N=1,KMAX
1059      NO=NUM(N)
1060      IS=1
1061      IF(NO.LT.0)IS=-1
1062      NO=NO*IS
1063      IF(JFIELD(N).GT.9)GOTO 110
1064      C      LAND GRID SQUARES
1065      DO 100 I=1,MAXDIG
1066      II=I+MAXDIG*(N-1)
1067      ANOM(II)=CHAR(1)
1068      IF(I.EQ.MAXDIG)ANUM(II)=CHAR(5)
1069      100 CONTINUE
1070      GOTO 200
1071      C      NUMBERS TOO LARGE: PRINT *****
1072      110 IF(NO.LE.1,IMAX(MAXDIG).AND.IS.NO.GT.
1073      1,-IMAX(MAXDIG)/10)GOTO 130
1074      DO 120 I=1,MAXDIG
1075      II=I+MAXDIG*(N-1)
1076      120 ANUM(II)=CHAR(4)
1077      GOTO 200
1078      C      WATER GRID SQUARES: FIRST INSERT BLANKS
1079      130 DO 140 I=1,MAXDIG
1080      II=I+MAXDIG*(N-1)
1081      140 ANUM(II)=CHAR(1)
1082      C      INSPECT GRIDS FROM LEFT TO RIGHT
1083      DO 150 J=1,MAXDIG
1084      I=MAXDIG+1-J
1085      II=I+MAXDIG*(N-1)
1086      IDIG=(NO-10***(MAXDIG+I-1)*(NO/(10***(MAXDIG+I-1))))
1087      1 /10***(MAXDIG-1)
1088      ANUM(II)=DIGIT(IDIG)
1089      C      LEAD BLANKS
1090      IF(NO.GT.10**3)IOR.J.EQ.MAXDIG)GOTO 150
1091      ANUM(II)=CHAR(1)
1092      C      SIGN CHARACTER
1093      IF(IS,LT,1)ANUM(II)=CHAR(3)
1094      GOTO 200
1095      150 CONTINUE
1096      200 CONTINUE
1097      RETURN
1098      END
1099 C-----+
1100 C-----+
1101 C-----+
1102      SUBROUTINE PRNCON(INDEX0)
1103
1104 C      PURPOSE - TO PRINT MOST OF THE RUN PARAMETERS IN A COMPACT FOR
1105 C      VARIABLES -
1106 C      INDEX0 = PARAMETER TO SPECIFY PRINTOUT OF SECONDARY
1107 C      VARIABLES LIKE DEPTHS (0=NO, 1=YES).  USUALLY
1108 C      INDEX0=1 AT START OF RUN, 0 LATER.
1109      INCLUDE 'COMM20.FOR'
1110      COMMON/A1/DTIMAX
1111      DIMENSION FRICH(10),ERR(10),FRT(10),NUM(101)
1112      DIMENSION JFIELD(40),ANUM(120)
1113      CHARACTER*2 TITLY(3)
1114      CHARACTER*2 CHNUM(2)
1115      CHARACTER*1 ANUM
1116      DATA CHNUM/'I. ','VI',TITLY/'D.', ' ', ' '
1117      JN=1
1118      IF(INDEX0.EQ.0)JN=2
1119      WRITE(10,70)VERS
1120      70 FORMAT(1X,'MECCA2 (MODEL FOR ESTUARINE AND COASTAL CIRCULATION',
1121      1 ' ASSESSMENT)',VERSION,'F5.2')
1122      WRITE(10,65)CHNUM(JN)
1123      65 FORMAT(1X,A2,' MODEL CONSTANTS',60(1H-))
1124      C      INPUT FILE NAMES
1125      IF(INDEX0.EQ.0)GOTO 85
1126      WRITE(10,80)FGE0
1127      80 FORMAT(//1X,'A. INPUT GEOGRAPHY FILE:',A40)
1128      85 CONTINUE
1129      C      PRINT OUT INTERNAL (SET IN THE CODE) PARAMETERS
1130      WRITE(10,90)AG,ALV,CPAIR,CWATER,
1131      1,E,EPSON,IO,LSCR,OMEGA,PA,PI,RHOA,RHOB,SB,SOLAR,VONKAR
1132      90 FORMAT(//1X,'B. INTERNAL PARAMETER VALUES:',/, 'AG=',F6.3,', ALV=',
1133      1,E10.4,', CPAIR=',F6.1,', CRATER=',F6.1,', E=',E10.4,/, 'AV00=',
1134      3,'EPSLON=',F5.3,', IO=',I2,', ISCR=',I2,', OMEGA=',E9.3,', PA=',
1135      4,F8.1,', PI=',F9.7,', RHOAE=',F5.3,/, 'RHOW=',F6.1,', SB=',E9.3,
1136      5,'SOLAR=',F6.1,', VONKAR=',F4.2)
1137      C      INPUT DATA VALUES
1138      WRITE(10,100)
1139      1 AHOO,AH,ALB,AVO0,AVO,BSNANG,BSNLAT,BSNLON,CAH,
1140      2 CDRGWS,CDBW1,CDBW2,CDR1,CDR2,CLOUD,CRO,(CRICH(I),I=1,8),
1141      3 DL,DTE,DVO,DVO0,D10FCT,DHAA,HRO,HRCNC,HRMAX,HROUT,
1142      4 HRROUT,IBETAA,IBETAB,IBETC,ICOR,ICOPL,ICPOS,IEXTRN,IGPH,
1143      5 IHEAT,IHVISC,INTER,ITOPV,IVISC,(JPRNT(I),I=1,13),KOCNHC,
1144      7 KONCEN,LAYS,MCOR,MMAX,NCOR,NMAX,NUMCR,NUMRIV,NUMBXY
1145      100 FORMAT(//1X,'C. INPUT DATA :',/, '1, C.1 CONFIGURATION DATA:',/,
1146      1,'AH00=','F5.1,', 'AH0=','F5.1,', 'ALB=','F4.2,
1147      2,'AV00=','F7.5,', 'AV0=','F7.5,', 'BSNANG=','F7.3,/, 'BSNLAT=','F7.3,
1148      3,'BSNLON','F8.3,', 'CAH=','F5.3,', 'CDRGWS=','F6.4,', 'CDWB1=','F6.4,
1149      4,'CDWB2=','F6.4,/, 'CDR1=','F8.6,', 'CLOUD=','F4.2,
1150      5,'CN=','F4.2)
1151      5 FORMAT(1X,E10.4,/, 'CRICH2=',E10.4,', 'CRICH3=',E10.4,', 'CRICH4=',
1152      6,E10.4,', 'CRICH5=',E10.4,', 'CRICH6=',E10.4,', 'CRICH7=',E10.4,
1153      7,'CRICH8=','E10.4,', 'DL=','F8.1,', 'DTE=','F8.2,/, 'DV00=','F8.6,
1154      7,'DVO=','F7.5,', 'DVO=','F7.5,', 'D10FCT=','F4.1,', 'DHAA=','F6.2,
1155      8,'HRO=','F6.1,', 'HRCNC=','F6.1,/, 'HRMAX=','F6.1,', 'HROUT=','F6.2,
1156      9,'HRROUT=','F6.1,', 'IBETAA=','I1,', 'IBETAB=','I1,', 'IBETC=','I1,
1157      1,'IBOTV=','I1,', 'ICOR=','I1,', 'ICOPL=','I1,', 'ICPOS=','I1,
1158      2,'IEXTRN=','I1,', 'I2,', 'IHEAT=','I2,', 'IHVISC=','I2,
1159      3,'INTER=','I2,', 'ITOPV=','I1,', 'IVISC=','I1,
1160      2,'JPRNT=','I132,', 'KOCNHC=','I2,
1161      3,'KONCEN=','I1,', 'LAYS=','I2,', 'MCOR=','I3,/, 'MMAX=','I3,', 'NCOR=',
1162      4,'I3,', 'NAME=','I3,', 'NUMBC=','I2,
1163      7,'NUMRIV=','I2,', 'NUMBX=','I4)
1164      C      FORMAT VARIABLES
1165      IF(INDEX0.EQ.0)GOTO 245
1166      WRITE(10,180),(JPRNT(I),I=1,13),KPRNT1,KPRNT2,NPRMN
1167      180 FORMAT(//1X,'C.2 PRINT VARIABLES:',/, '1, JPRNT=','1312,
1168      1 IF(NPRMN,GT,1)NPRMN,IPRMN(J),J=1,NPRMN)
1169      190 FORMAT(1X,IPRMN=190)
1170      190 IF(1SLICE.EQ.0)GOTO 220
1171      WRITE(10,195)1SLICE
1172      195 FORMAT(1X,'SLICE PRINT SECTIONS: 1SLICE=',I2)
1173      DO 200 I=1,1SLICE
1174      200 JS=1SLICE(I)
1175      245 CONTINUE
1176      DO 197 J=1,JS
1177      197 NUM(J)=100*MSLICE(J,I)+NSLICE(J,I)
1178      200 WRITE(10,210)JS,(NUM(J),J=1,JS)
1179      210 FORMAT(1X,'JSLICE=',I2,', M_NSLICE=',5I7,/,10I7,/,10I7)
1180      210 WRITE(10,212)IGPH,NSTGPB,IGPHOP
1181      212 FORMAT(1X,'VARIABLES SAVED FOR GRAPHING: IGPH=',I2,
1182      1,'NSTGPB=','I3,', 'IGPHOP=','i1)
1183      I=1
1184      IF(IGPH,GT,0)WRITE(10,216)(I,LGHF(I),MGHF(I),NGHF(I),
1185      1 ITYP(I),PTILE(ITYP(I)),I=1,IGPH)
1186      216 FORMAT(1IX,I2,:'L='I2,'M='I3,'N='I3,'ITYP='I2,
1187      1,'A10,'))
1188      220 CONTINUE
1189      245 CONTINUE
1190      C      WRITE THE SECONDARY PARAMETERS
1191      AHRMAX=DL*(8.*DTE)
1192      DTMAX=DL*(TIME STEP FOR EXTERNAL, INTERNAL MODES
1193      DTEMAX=DL/SQRT(DMAX*AG)
1194      DTMAX=DL/SQRT(DMAX*AG)
1195      DMAX=MAX(ANMAX,ANMIN)(ISPLIT)
1196      WRITE(10,102)COR,DENRA,DFDM,DFDN,DMAX,DMIN,DO,
1197      1,DTI,IHR,KRIVBC,NBCEL,NCELL,NSTET,NSTI,NSTIMX,RAD
1198      102 FORMAT(//1X,'D. PARAMETERS COMPUTED FROM INPUT DATA:',/,
1199      1,'COR=','F8.6,', 'DENRA=','F7.5,', 'DFDM=','B8.2,
1200      2,'DFDN=','E8.2,', 'DMAX=','F8.2,', 'DMIN=','F8.2,', 'DQ=','F5.3,
1201      3,'DTI=','F7.1,', 'IHR=','I3,', 'KRIVBC=','I1,/',
1202      4,'NBCEL=','I3,', 'NCELL=','I6,
1203      5,'NSTET=','I9,', 'NSTI=','I9,', 'NSTIMX=','I5,', 'RAD=','F10.8)
1204      SUM=0.0
1205      DO 115 L=L1,LBOT
1206      115 SUM=SUM+CI(L)
1207      WRITE(10,120)(CI(L),L=1,2),SUM1
1208      120 FORMAT(1X,'CI(1)=',F6.4,', CI(2)=',F6.4,', SUM OF LBOT CIS= ',F4.2)
1209      C
1210      IF(INDEX0.EQ.0)GOTO 300
1211      DTEMAX=DL/SQRT(2.*DMAX*AG)
1212      DTIMAX=(DMIN*DQ)*2/(2.*AV00)
1213      F=3600./DTI
1214      WRITE(10,160)DTEMAX,DTIMAX,F
1215      160 FORMAT(1X,'E. CHESTS')
1216      2 2X,10*STEPES,DEMAX,DL/SQRT(DMAX*AG)= ,F11.2,/,
1217      3 13X,DTIMAX,(DMIN*DQ)*2/(8*AV00)= ,F10.2,/
1218      4 13X,IHR,(STEPES PER HOUR)=3600/DTI= ,F12.6)
1219      166 WRITE(10,166)KONCEN,ICOPL,IBETAP
1220      166 FORMAT(2X,'DENSITY: KONCEN=',I1,', ICOPL=',I1,', IBETAP=',I1)
1221      170 CONTINUE
1222      300 CONTINUE
1223      700 CONTINUE
1224      RETURN
1225      END
1226      C
1227      C-----+
1228      C-----+
1229      SUBROUTINE PRNCGC(INDEX)
1230
1231      C      PURPOSE - TO PRINT MOST OF THE RUN PARAMETERS IN A COMPACT FOR
1232      C      VARIABLES -

```

```

1233 C      INDEX = PARAMETER TO SPECIFY PRINTOUT OF SECONDARY
1234 C      VARIABLES LIKE DEPTHS (0=NO, 1=YES). USUALLY
1235 C      INDEX=1 AT START OF RUN, 0 LATER.
1236 INCLUDE 'COMM20.FOR'
1237 DIMENSION TITLX(3),DAVG(NPMSIZ),NUMN(NPMSIZ),
1238 1 NUMN(NPMSIZ),NMBRM(NSIZE),NMBRN(NSIZE),ICELL(5),JFIELD(NSIZE)
1239 CHARACTER*10 TITLX
1240 CHARACTER*2 TITLY(3)
1241 CHARACTER*ANUM(3*NSIZE)
1242 DATA TITLY/'D.,',',','/
1243 DATA TITLX/'AREA*100. ','BX*100. ','BY*100. '/
1244 DATA ICEL5/*0/
1245 C      PRINT INPUT DATA
1246 IF(INDEX,NE.1)RETURN
1247 WRITE(IO,100)
1248 100 FORMAT(//,1X,'II. TIME-VARIABLE INPUT',55(1H-),/1X,
1249 1 'A. REFERENCE DATE ')
1250 C      ENVIRONMENTAL DATA
1251 110 WRITE(IO,120)IYEAR,MONTH,IDAY,IHOUR,IMIN,UTO
1252 120 FORMAT(IX,'STARTING DATE: IYEAR=',I4,' MONTH=',I2,' IDAY=',I2,
1253 1 ' IHOUR=',I4,' IMIN=',I3,' UTCO=',F10.4)
1254 C      PRINT WATER LEVEL DATA (TIDES)
1255 WRITE(IO,150)NSIGW
1256 150 FORMAT(1X,'B. OCEAN WATER LEVEL DATA: NSIGT=',I2)
1257 IF(NSIGT,GT,0)THEN
1258 DO I=1,2
1259 WRITE(IO,180)I,YTID(I),DTID(I),(TLEV(I,N),N=1,NSIGT)
1260 180 FORMAT(5X,'RECORD',I2,F8.1,F10.4,20F10.2)
1261 ENDO
1262 ENDIF
1263 C      PRINT WINDS AND AIR TEMPERATURE
1264 WRITE(IO,170)NSIGW
1265 170 FORMAT(1X,'C. WIND DATA: NSIGW=',I2)
1266 C      PRINT RIVER FLOW
1267 WRITE(IO,190)NSIGR
1268 190 FORMAT(1X,'D. RIVER FLOW RATE DATA: NSIGR=',I2)
1269 IF(NSIGR,GT,0)THEN
1270 DO I=1,2
1271 WRITE(IO,180)I,YRIV(I),(DRIV(I,N),N=1,NSIGR)
1272 ENDO
1273 ENDIF
1274 C      PRINT OCEAN BOUNDARY SALINITY
1275 WRITE(IO,210)NSIGS
1276 210 FORMAT(//,1X,'E. OCEANIC BOUNDARY SALINITY: NSIGS=',I2)
1277 IF(NSIGS,GT,0)THEN
1278 DO I=1,2
1279 WRITE(IO,180)I,YSLA(I),DSAL(I),(SALOCN(I,N),N=1,NSIGS)
1280 ENDO
1281 ENDIF
1282 C      PRINT OCEAN BOUNDARY TEMPERATURE
1283 WRITE(IO,230)NSIGTO
1284 230 FORMAT(//,1X,'F. OCEANIC BOUNDARY TEMPERATURE: NSIGTO=',I2)
1285 IF(NSIGTO,GT,0)THEN
1286 DO I=1,2
1287 WRITE(IO,180)I,YOTP(I),DOTP(I),(TMPOCN(I,N),N=1,NSIGTO)
1288 ENDO
1289 ENDIF
1290 C      PRINT RIVER BOUNDARY TEMPERATURE
1291 WRITE(IO,250)NSIGRT
1292 250 FORMAT(//,1X,'G. RIVER BOUNDARY TEMPERATURE: NSIGRT=',I2)
1293 IF(NSIGRT,GT,0)THEN
1294 DO I=1,2
1295 WRITE(IO,180)I,YRVT(I),DRV(T(I),(TRIV(I,N),N=1,NSIGRT)
1296 ENDO
1297 ENDIF
1298 C      PRINT ADDITIONAL MET DATA
1299 WRITE(IO,260)NSIGM
1300 260 FORMAT(//,1X,'H. ADDITIONAL MET DATA: NSIGM=',I2)
1301 IF(NSIGM,GT,0)THEN
1302 DO I=1,2
1303 WRITE(IO,180)I,YMET(I),DMET(I),(VMET(I,N),N=1,NSIGM)
1304 ENDO
1305 ENDIF
1306
1307 C      BOUNDARY LOCATIONS
1308 IF(NUMBOC+NUMBIV,GT,0)WRITE(IO,410)
1309 410 FORMAT(//,1X,'BOUNDARY SPECIFICATIONS IN GEO FILE')
1310 IF(NUMBOC,GT,0)THEN
1311 DO 420 N=1,NUMBOC
1312 420 WRITE(IO,430)N,MB1(N),MB2(N),NB1(N),NB2(N),ITPO(N),JTP0(N),
1313 1 ISET1(N),ISET2(N)
1314 430 FORMAT(1X,'OCEAN BND. ',I2,' MB1,2=',I2,I4,' NB1,2=',I2,I4,
1315 1 ' ITPO=',I2,' JTP0,',I2,' ISET1,2=',I2,I4)
1316 ENDIF
1317 IF(NUMBIV,GT,0)THEN
1318 DO 440 N=1,NUMBIV
1319 440 WRITE(IO,450)N,MRI1(N),MR2(N),NR1(N),NR2(N),ITPR(N),JTPR(N),
1320 1 ISERT(N)
1321 450 FORMAT(1X,'RIVER BND. ',I2,' MRI,2=',I2,I4,' NR1,2=',I2,I4,
1322 1 ' JTPR,',I2,' ITPR,',I2,' ISERT=',I2)
1323 ENDIF
1324 C      CATEGORIES OF IFIELD VARIABLES
1325 480 DO 490 N=1,NMAX
1326 490 NMBRN(N)=N
1327 DO 492 M=1,MMAX
1328 492 NMBRM(M)=M
1329 DO 493 M=1,MMAX
1330 493 N=1,NMAX
1331 IF(IFIELD(N,M),LT,10)ICELL(1)=ICELL(1)+1
1332 IF(IFIELD(N,M)/10.EQ.1)ICELL(2)=ICELL(2)+1
1333 IF(IFIELD(N,M)/10.EQ.2)ICELL(3)=ICELL(3)+1
1334 IF(IFIELD(N,M)/10.EQ.KOCNBC)ICELL(4)=ICELL(4)+1
1335 IF(IFIELD(N,M)/10.EQ.KOCNBC+1)ICELL(5)=ICELL(5)+1
1336 495 CONTINUE
1337 ISUM=0
1338 DO 496 I=1,5
1339 496 ISUM=ISUM+ICELL(I)
1340 JSUM=NMAX*MMAX
1341 C      PRINT IFIELD
1342 KPI=1,INE=1
1343 WRITE(IO,500)KOCNBC,KP1,BSNANG,BSNLAT,BSNLON,DL,MCOR,NCOR
1344 500 FORMAT(//,1X,'I. CELL CODECS: ',/4X,
1345 1 'X. ',A,' II. CELL CODECS: ',/4X,
1346 2 ' LAND= ',TRIANGULAR-1,WATER=20,OCEAN BND='I1,'0 RIVER BND=',
1347 3 'I1,'0',/1X,' WATER & NO X-FLOW=21',
1348 4 '4X,' WATER & NO Y-FLOW=22 WATER & NO X- OR Y-FLOW=23',//,
1349 5 'IX,' BSNANG='F8.3,' BSNLAT='F8.3,' BSNLON='F8.3,' DL=',
1350 6 'F9.2,' MCOR='I2,' NCOR='I2')
1351 WRITE(IO,510)(ICELL(J),J=1,5),ISUM,JSUM
1352 510 FORMAT(//,1X,'CELL COUNTS: LAND=',I6,/2X,'WATER:TRIANGULAR=',I6,//,
1353 1 '6X,'WATER:SQUARE=',I6,/9X,'OCEAN BND=',I6,/9X,'RIVER BND=',I6,
1354 2 '/10X,'CELL SUM=',I6,' VS. NMAX*MMAX=',I6)
1355 C      PRINT DEPTHS
1356 C      PRINT DEPTHS
1357 530 NPERL=24
1358 KMAX=1+(NMAX-1)/NPERL
1359 WRITE(IO,531)
1360 531 FORMAT(//,1X,'C. DEPTHS (M) AT MSL')
1361 DO 563 K=1,KMAX
1362 N1=1+(K-1)*NPERL
1363 N2=MINO(NMAX,N1+NPERL-1)
1364 WRITE(IO,532)(NMAX,N1+NPERL-1)
1365 532 FORMAT(2X,' M ',N1,I2,30I3)
1366 DO 550 M=1,MMAX
1367 DO 540 N=N1,N2
1368 NUMN(N)=-
1369 540 IF(IFIELD(N,M).GT.0)NUMN(N)=D(N,M)
1370 WRITE(IO,560)(NMAX,N1,N2)
1371 560 FORMAT(1X,I3,2X,30I3)
1372 550 CONTINUE
1373 563 CONTINUE
1374
1375 C      COMPUTE AVERAGE DEPTHS
1376 DO 570 M=1,MMAX
1377 NUMN(M)=M
1378 NN=0
1379 SUM=0.
1380 DO 565 N=1,NMAX
1381 IF((N,M).LT.0)GOTO 565
1382 NN=N+1
1383 SUM=SUM+D(N,M)
1384 565 CONTINUE
1385 570 DAVG(M)=SUM/(E+FLOAT(NN))
1386 WRITE(IO,580)
1387 580 FORMAT(//,1X,'MEAN OF NON-ZERO DEPTHS')
1388 NPL=5
1389 KMAX=1+(NMAX-1)/NPL
1390 DO 600 K=1,KMAX
1391 M1=1+(K-1)*NPL
1392 M2=MINO(NMAX,M1+NPL-1)
1393 595 WRITE(IO,591)(NUMN(M),DAVG(M),M=M1,M2)
1394 591 FORMAT(1X,M,D=',5(I3,F8.2,2,2))
1395 591 FORMAT(1X,M,D=',5(I3,F8.2,2,2))
1396 600 CONTINUE
1397
1398 C      VARIABLE WIDTHS
1399 IF(NUBMXY.LE.0)GOTO 660
1400 N3MAX=3*NMAX
1401 DO 650 J=1,3
1402 WRITE(IO,610)TITLX(J),TITLX(J)
1403 610 FORMAT(1X,A2,' VARIABLE-WIDTH PARAMETERS: ',A10)
1404 NPL=25 ! number per line
1405 KMAX=1+(NMAX-1)/NPL
1406 DO 640 K=1,KMAX
1407 N1=1+NPL*(K-1)
1408 N2=MINO(NMAX,N1+NPL-1)
1409 WRITE(IO,612)(NMAX,N1+NPL-1)
1410 612 FORMAT(4X,40I3)
1411 DO 630 M=1,MMAX
1412 DO 620 N=N1,N2
1413 IF(J.EQ.1)F1=AREA(N,M)
1414 IF(J.EQ.2)F1=BX(N,M)
1415 IF(J.EQ.3)F1=BY(N,M)
1416 JFIELD(N)=JFIELD(N,M)
1417 620 NUMN(N)=F1*100.
1418 CALL PRNCHR(3,NUMN,JFIELD,ANUM,NMAX)
1419 N1A=1+3*(N1-1)
1420 N2A=MINO(3*NMAX,N1A-1+3*NPL)
1421 WRITE(IO,635)(M,(ANUM(N),N=N1A,N2A)
1422 635 FORMAT(1X,I3,120A1,3/(5X,120A1))
1423 630 CONTINUE ! end mmax
1424 640 CONTINUE ! end k
1425 650 CONTINUE ! end j
1426 660 CONTINUE
1427 C      PRINT FLAGS
1428 DO 670 N=1,NSIZE
1429 670 NUMN(N)=N
1430 C      WRITE OUT THE TAGS
1431 NX=MAX(1,NCOL,NROW)
1432 WRITE(IO,675)
1433 675 FORMAT(//,1X,'E. ROW/COLUMN FLAGS:',/6X,'B.C. FOR IL/IR: 0=WATER',
1434 1 ' LEVEL, 1=FLOWRATE, 2=ZERO FLOW',/12X,2('ICOL',14X,'IROW',
1435 2 ,18X,'/2X,2('I',4X,'N MA MB IL IR',4X,'M NA NB IL IR',3X))
1436 2 ,18X,'/2X,2('I',4X,'N MA MB L R',4X,'M NA NB L R',3X))
1437 NXM=1+(NX-1)/2
1438 DO 680 N=1,NXM
1439 M=N+NXM
1440 680 WRITE(IO,690)(N,(ICOL(K,N),K=1,5),(IROW(K,N),K=1,5),M,
1441 1 (ICOL(K,M),K=1,5),(IROW(K,M),K=1,5)
1442 690 FORMAT(1X,2(I3,:1,314,212,2,X,314,212,2X))
1443 C      PRINT OUT NEW IFIELD (WITH BARRIERS)
1444 NPL=30 ! number per line
1445 KMAX=1+(NMAX-1)/NPL
1446 DO 730 K=1,KMAX
1447 N1=1+NPL*(K-1)
1448 N2=MINO(NMAX,N1+NPL-1)
1449 WRITE(IO,692)(NUMN(I),I=N1,N2)
1450 692 FORMAT(//,1X,'F. MODIFIED IFIELD',/1X,' M NA NB ',40I2,
1451 1 ' 3/(10X,40I2))
1452 DO 700 M=1,MMAX
1453 NA=NUM(BM)/1000
1454 NB=MOD(NAB(M),1000)
1455 700 WRITE(IO,710)M,NA,NB,(IFIELD(I,M),I=N1,N2)
1456 710 FORMAT(1X,I3,214,2X,40I2,3/(14X,40I2))
1457 730 CONTINUE
1458
1459 DO 780 K=1,KMAX
1460 N1=1+NPL*(K-1)
1461 N2=MINO(NMAX,N1+NPL-1)
1462 WRITE(IO,750)(NUMN(I),I=N1,N2)
1463 750 FORMAT(//,1X,'MFLUX',/6X,40I2)
1464 DO 760 M=1,MMAX
1465 760 WRITE(IO,770)(M,(MFLUX(I,M),I=N1,N2)
1466 770 FORMAT(1X,I3,2X,40I2)
1467 780 CONTINUE
1468
1469 DO 820 K=1,KMAX
1470 N1=1+NPL*(K-1)
1471 N2=MINO(NMAX,N1+NPL-1)
1472 WRITE(IO,800)(NUMN(I),I=N1,N2)
1473 800 FORMAT(//,1X,'NFLUX',/6X,40I2)
1474 DO 810 M=1,MMAX
1475 810 WRITE(IO,770)(M,(NFLUX(I,M),I=N1,N2)
1476 820 CONTINUE
1477 RETURN
1478 END
1479 C
1480 C-----SUBROUTINE PSLICE
1481 C-----JANUARY 1985 K. W. HESS MEAD VAX 11/750
1482 SUBROUTINE PSLICE
1483 C      PURPOSE - PRINT ALL CONCENTRATIONS OVER DEPTH AT ALL GRID CELLS
1484 C

```

```

1485 C          ALONG A SLICE OF THE DOMAIN. UP TO
1486 C          5 SLICES ALLOWED, EACH WITH UP TO 10 POINTS.
1487 C
1488 C          VARIABLES
1489 C          ISLICE = NO. OF SLICES
1490 C          ITYPE = TYPE OF VARIABLE
1491 C          JSLICE(I) = NO. OF POINTS IN I-TH SLICE
1492 C          MSLICE(I,J) = J-TH VALUE OF THE M-GRID POINT ON THE I-TH SLICE
1493 C          INCLUDE 'COMM20.FOR'
1494 C          DIMENSION NO(NM2SIZ),NDAT(LSIZE,NM2SIZ),II(NM2SIZ),
1495 C          PSE(NM2SIZ),DP(NM2SIZ),F(LSIZE,NM2SIZ),ITYPE(8)
1496 C          DATA ITYPE/4,5,6,9,10,22,15,16/ ! UT,VT,W, S,T, AH3,AV,DV
1497 C          DO 100 N=1,NM2SIZ
1498 C          II(N)=N
1499 C          ANUM(N)='====';
1500 C          100 BNUM(N)='----';
1501 C          DO 1000 N=1,ISLICE
1502 C          IF(ISLICE.LE.0)RETURN
1503 C          DO 100 N=1,ISLICE
1504 C          DO 300 ISL=1,ISLICE
1505 C          CONSTRUCT THE SLICE
1506 C          K=0
1507 C          DO 130 J=1,JSLICE(IS)-1
1508 C          NA=NSLICE(J,IS)
1509 C          MA=MSLICE(J,IS)
1510 C          NB=NSLICE(J+1,IS)
1511 C          MB=MSLICE(J+1,IS)
1512 C          GET DIRECTION (INCREASING OR DECREASING)
1513 C          MDIR=ISIGN(1,MB-MA)
1514 C          NDIR=ISIGN(1,NB-NA)
1515 C          IF(NB.EQ.NA)NDIR=0
1516 C          IF(MB.EQ.MA)MDIR=0
1517 C          LINE: save m, n, and depth
1518 C          NN=NA-NDIR
1519 C          MM=MA-MDIR
1520 C          110 MM=MM+MDIR
1521 C          NN=NN+NDIR
1522 C          IF(K.GT.1.AND.(MM.EQ.NO(K)).AND.NN.EQ.NO(K))GOTO 110
1523 C          K=K+1
1524 C          DP(K)=D(NN,MM)
1525 C          PSE(K)=SE(NN,MM)
1526 C          MO(K)=MM
1527 C          NO(N)=MM
1528 C          IF(MM.EQ.MB.AND.NN.EQ.NB)GOTO 130
1529 C          GOTO 110
1530 C          130 CONTINUE
1531 C          KMAX=K
1532 C          LOOP THRU THE VARIABLES
1533 C          DO 270 I=1,8
1534 C          LEVS=LBOT
1535 C          IF(IITYPE(IT).EQ.6)LEVS=LAYS
1536 C          IF(IITYPE(IT).GE.15.AND.IITYPE(IT).LE.17)LEVS=LAYS
1537 C          FMAX=0.
1538 C          DO 140 L=1,LEVS
1539 C          DO 140 K=1,KMAX
1540 C          F(L,K)=SELECT(IITYPE(IT),L,NO(K),MO(K))
1541 C          140 FMAX=AMAX1(ABS(F(L,K)),FMAX)
1542 C          scale
1543 C          I1=0
1544 C          IF(FMAX.GT.0)I1=INT(ALOG10(FMAX)+10.)-13
1545 C          SCL=10.*I1
1546 C          DO 145 L=1,LEVS
1547 C          DO 145 K=1,KMAX
1548 C          145 NDAT(L,K)=NINT(F(L,K)/SCL)
1549 C          loop thru sets
1550 C          KPERL=4
1551 C          KSET=(KMAX-1)/KPERL+1
1552 C          DO KS=1,KSET
1553 C          WRITE OUT THE HEADER LINE
1554 C          IF(KS.EQ.1)WRITE(IO,150)PTITLE(IITYPE(IT)),SCL,UT,FMAX
1555 C          150 FORMAT(1X,'SECTION : ',A10,', DIVIDED BY ',1PE9.3,', UT=',
1556 C          1,0PE10.4,' FMAX=',E10.4)
1557 C          K1=1+KPERL(KS-1)
1558 C          K2=MINO(KMAX,K1+KPERL-1)
1559 C          WRITE(IO,152)(I1(K),K=K1,K2)
1560 C          152 FORMAT(1X,'K',14I5)
1561 C          WRITE(IO,160)(NO(K),K=K1,K2)
1562 C          160 FORMAT(1X,'N',14I5)
1563 C          WRITE(IO,170)(MO(K),K=K1,K2)
1564 C          170 FORMAT(1X,'M',14I5)
1565 C          depths
1566 C          WRITE(IO,210)(DP(K),K=K1,K2)
1567 C          210 FORMAT(2X,' D=',14F5.1)
1568 C          WRITE(IO,220)(PSE(K),K=K1,K2)
1569 C          220 FORMAT(2X,' SE=',14F5.2)
1570 C          WRITE(IO,240)(BNUM(K),K=K1,K2)
1571 C          240 FORMAT(6X,14A5)
1572 C          RUN DOWN THE COLUMN
1573 C          DO 250 I=1,1,LEVS
1574 C          250 WRITE(IO,250)(NDAT(L,K),K=K1,K2)
1575 C          260 FORMAT(1X,'L',12,1X,14I5)
1576 C          WRITE(IO,265)ANUM(1),(ANUM(K),K=K1,K2)
1577 C          265 FORMAT(X,15A5)
1578 C          ENDDO ! end set
1579 C          270 CONTINUE ! end type
1580 C          300 CONTINUE ! end slice
1581 C          RETURN
1582 C          END
1583 C
1584 C-----SUBROUTINE PGRAPH
1585 C          OCTOBER 1984 K. HESS MEAD VAX 11/750
1586 C          PURPOSE - TO SAVE VARIABLES FOR LATER GRAPHING
1587 C          VARIABLES -
1588 C          IO1,IO2 = CHANNEL NUMBERS FOR OUTPUT (20,21).
1589 C          (SET IN SUB. INITs)
1590 C          INCLUDE 'COMM20.FOR'
1591 C          DIMENSION XGR(NDGPH),NUM(NDGPH)
1592 C          DATA PTITLE/
1593 C          1 'SE      ','UE      ','VE      ','UTotal   ','Vtotal   ',
1594 C          2 'W      '(m/s),'TSX*1.E-7 ','TSY*1.E-7 ','IS(PFT)  ','IT(C)  ',
1595 C          3 'AH/H    ','THETA1  ','THETA2  ','THETA3  ','AV      ',
1596 C          4 'DV      ','RI      ','WX      ','WY      ','UHOLD/H  ',
1597 C          5 'VHOLD/H ','AH3/H   ','          ','          ','          '
1598 C          CHECK FOR NUMBER OF GRAPHING VARIABLES
1599 C          IF(IGPH.LE.0)RETURN
1600 C          CHECK FOR WRITE:
1601 C          IF(.NOT.(NSTI.EQ.0.OR.NSTI.EQ.NSTIMX.OR.NSTI.EQ.NSTGPH*(NSTI/
1602 C          1 NSTGPH)))GOTO 210
1603 C          STORE PRESENT VALUES IN ARRAY "XGR"
1604 C          DO 110 I=1,IGPH
1605 C          110 IF(I.GT.NDGP)GOTO 110
1606 C          DO 110 I=1,IGPH
1607 C          110 IF(I.GT.NDGP)GOTO 110
1608 C          NUM(I)=I
1609 C          N=NGPH(I)
1610 C          M=MGPH(I)
1611 C          L=LGPH(I)
1612 C          XGR(I)=SELECT(ITYP(I),L,N,M)
1613 C          110 CONTINUE
1614 C          LOOP THRU VARIABLES
1615 C          NUMPLN=IGPH
1616 C          KMAX=1+(IGPH-1)/NUMPLN
1617 C          DO 200 K=1,KMAX
1618 C          J1=1+NUMPLN*(K-1)
1619 C          J2=MINO(J1+NUMPLN-1,IGPH)
1620 C          WRITE HEADER: IGXX=FORMAT, IV=INTVEL, NKOLS=NO. COLUMNS HERE
1621 C          IF(NSTI.GT.0.OR.IGPHOP.EQ.0)GOTO 125
1622 C          125 NKOLS=1+J2-J1
1623 C          WRITE(LUGRF,112)CTITLE,IGPH,NSTIMX,NSTGPH,DTI,YEAR
1624 C          112 FORMAT(8A10,/,1X,3I7,F9.3,2F9.3)
1625 C          115 WRITE(LUGRF,120)J,PTITLE(ITYP(J)),LGPH(J),MGPH(J),NGPH(J)
1626 C          120 FORMAT(2X,'COLUMN ',I3,' HAS ',A10,6H AT L=,I2,3H M=,I3,3H N=,I3)
1627 C          125 WRITE(LUGRF,124)(NUM(I),I=J1,J2)
1628 C          124 FORMAT(4X,'YEAR UT,(0,4011)
1629 C          125 WRITE(LUGRF,130)YEAR,UT,(XGR(J),J=J1,J2)
1630 C          129 FORMAT(1X,F6.0,F9.4,40(x,F10.4))
1631 C          130 WRITE NUMBERS AT END
1632 C          134 C-----FORMAT(1X,F6.0,F9.4,40(x,F10.4))
1633 C          135 200 IF(NST.EQ.NSTIMX.AND.IGPHOP.EQ.1)WRITE(LUGRF,124)(NUM(I),
1634 C          136 I=J1,J2)
1635 C          137 200 RETURN
1636 C          138 END
1637 C-----END
1638 C-----FUNCTION SELECT(ITYP,L,N,M)
1639 C-----SELECT A VARIABLE BASED ON THE GRAPHING TABLE VALUES
1640 C-----IGPH RANGES FROM 1 TO 21
1641 C-----INCLUDE 'COMM20.FOR'
1642 C-----SELECT=0.0
1643 C-----MP=MINO(M1,MMAX)
1644 C-----NP=MINO(N1,NMAX)
1645 C-----GOTO(100,110,120,130,140,150,160,170,180,190,200,210,
1646 C-----1 220,230,240,250,260,270,280,290,300,310),ITYP
1647 C-----GOTO 400
1648 C-----SELECT=SE(N,M) ! 1
1649 C-----GOTO 400
1650 C-----SELECT=UE(N,M) ! 2
1651 C-----GOTO 400
1652 C-----SELECT=VE(N,M) ! 3
1653 C-----GOTO 400
1654 C-----SELECT=UE(N,M) ! 4
1655 C-----GOTO 400
1656 C-----SELECT=VE(N,M) ! 5
1657 C-----GOTO 400
1658 C-----SELECT=UE(N,M)+U(L,N,M)
1659 C-----GOTO 400
1660 C-----SELECT=VE(N,M)+V(L,N,M) ! 6
1661 C-----GOTO 400
1662 C-----SELECT=W(L,N,M)
1663 C-----GOTO 400
1664 C-----SELECT=TSX(N,M)*1.E+7 ! 7
1665 C-----GOTO 400
1666 C-----SELECT=TSY(N,M)*1.E+7
1667 C-----GOTO 400
1668 C-----SELECT=S(L,N,M)
1669 C-----GOTO 400
1670 C-----SELECT=T(L,N,M) ! 10
1671 C-----GOTO 400
1672 C-----SELECT=AH(N,M)/(D(N,M)+SE(N,M)+E)
1673 C-----GOTO 400
1674 C-----SELECT=THETA1(N,M) ! 12
1675 C-----GOTO 400
1676 C-----SELECT=THETA2(N,M)
1677 C-----GOTO 400
1678 C-----SELECT=THETA3(N,M)
1679 C-----GOTO 400
1680 C-----SELECT=AV(L,N,M)*1.E+4 ! 15
1681 C-----GOTO 400
1682 C-----SELECT=DV(L,N,M)*1.E+4
1683 C-----GOTO 400
1684 C-----SELECT=R1(L,N,M)
1685 C-----GOTO 400
1686 C-----SELECT=WX(N,M) ! 18
1687 C-----GOTO 400
1688 C-----SELECT=WY(N,M)
1689 C-----GOTO 400
1690 C-----SELECT=UHOLD(N,M)/(D(N,M)+SE(N,M)+D(N,MP)+SE(N,MP)) ! 20
1691 C-----GOTO 400
1692 C-----SELECT=VHOLD(N,M)/(D(N,M)+SE(N,M)+D(NP,M)+SE(NP,M)) ! 21
1693 C-----GOTO 400
1694 C-----SELECT=AH3(L,N,M)/(D(N,M)+SE(N,M)+E) ! 22
1695 C-----400 CONTINUE
1696 C-----RETURN
1697 C-----END
1698 C-----MECCA FILE : EXMODE.FOR
1700 C-----SUBROUTINE EXMODE
1701 C-----JANUARY 1986 K.W. HESS MEAD VAX 11/750
1702 C-----PURPOSE - TO COMPUTE EXTERNAL-MODE FLOWRATES AND WATER LEVELS
1703 C-----FROM REVISED EQUATIONS FOR VARIABLE GRID WIDTH (TO
1704 C-----MODEL NARROW RIVER WIDTHS).
1705 C-----INCLUDE 'COMM20.FOR'
1706 C-----UPDATE THETA FUNCTIONS
1707 C-----SET THE BOUNDARY CONDITIONS
1708 C-----CALL THETAS
1709 C-----CALL BNDRY1
1710 C-----SET THE BOUNDARY CONDITIONS
1711 C-----CALL BNDRY1
1712 C-----UPDATE HORIZONTAL VISCOSITY COEFFICIENTS
1713 C-----STORE PREVIOUS WATER LEVELS FOR THE SALINITY AND TEMP. CALCS.
1714 C-----IF(NSTE.GT.1)GOTO 150
1715 C-----DO 130 N=1,NMAX
1716 C-----DO 130 M=1,MMAX
1717 C-----DO 130 L=1,LMAX
1718 C-----UHOLD(N,M)=UH(N,M)
1719 C-----VHOLD(N,M)=VH(N,M)
1720 C-----SOLD(N,M)=SE(N,M)
1721 C-----130 CONTINUE
1722 C-----COMPUTE VELOCITIES
1723 C-----CALL UHVF
1724 C-----RETURN
1725 C-----END
1726 C-----C-----SUBROUTINE BNDRY1
1727 C-----OCTOBER 1984 K. W. HESS MEAD VAX11/750
1728 C-----PURPOSE - TO SET THE EXTERNAL-MODE OPEN BOUNDARY WATER LEVEL OR
1729 C-----FLOWRATE CONDITIONS.
1730 C-----VARIABLES -
1731 C-----VAL = TIDAL ELEVATION/FLOWRATE AT ENDS OF OCEAN BOUNDARY
1732 C-----FT = TIME PAST PREVIOUS HI/LO (HR)
1733 C-----IS = GRID INCREMENT FOR RADIATION TO SET VELOCITY AT
1734 C-----1735 C-----1736 C-----
```

```

1737 C OUTSIDE EQUAL TO VELOCITY IN INTERIOR.
1738 C ITPO = DIRECTION OF OUTFLOW: 1=X, -1=-X, 2=Y, -2=-Y
1739 C JTPO = TYPE OF OCEAN BOUNDARY CONDITION
1740 C 1 : WATER LEVEL SPECIFICATION
1741 C 2 : TRANSPORT OUTFLOW
1742 C 3 : ORLANSKI RADIATION OUTFLOW (REQUIRES WL)
1743 C 4 : REIMANN INVARIANT RADIATION OUTFLOW (REQUIRES
1744 C ITPR = DIRECTION OF RIVER INFLOW: 1=X, -1=-X, 2=Y, -2=-Y
1745 C JTPR = RIVERINE CONDITION: 1=FLUME, 2=WATER FALLS
1746 C J = GRID INCREMENTS FOR TIDES TO SET VELOCITY AT
1747 C OUTSIDE EQUAL TO VELOCITY IN INTERIOR.
1748 C NUMOBC = NUMBER OF TIDAL BOUNDARIES
1749 C QRV = RIVER FLOWRATE (M**3/S)
1750 INCLUDE 'COMM20.FOR'
1751 DIMENSION FINAL(NDTID2),QRATE(NDRIV2)
1752 C1=DTE/DL
1753 ITEST=IPRNT1
1754 IF(NSTI.GT.10.AND.NSTE.GT.1) ITEST=0
1755 IF(ITEST.GT.0) WRITE(ISCR,100) UT,NSTI,NSTE,NUMOBC,NUMRIV
1756 100 FORMAT(1X,'BNDRY1_UT=',F10.4,' NSTI=',I6,' NSTE=',I2,
1757 1,' RAMPR_AMN1(1,0,CUMDAY),
1758 1,' RAMPR_AMN1(1,0,CUMDAY),
1759 1,' RAMPR_AMN1(1,0,CUMDAY),
1760 1,' NUMOBC,LE,0)GOTO 220
1761 C 1 LOOP TIDE OCEANIC BOUNDARIES
1762 DO 200 IB=1,NUMOBC
1763 C FIND THE LATEST VALUES FROM INPUT DATA
1764 CALL RR(YD,ISCR,LUTID,LENDTD,DTID,YTID,NSIGT,NDTID2,TDELEV,FINAL)
1765 F1=ISIGN(1,ITPO(IB))
1766 IUH=-1*(SIGN(1,ITPO(IB))+1)/2
1767 ISE=-ISIGN(1,ITPO(IB))
1768 IDIR=IABS(ITPO(IB))
1769 IF(ITEST.GT.0) WRITE(ISCR,110) IB,F1,ISE,IUH,DIR,JTP0(IB),FINAL(IB)
1770 110 FORMAT(1X,'IB=',I2,' F1,ISE,IUH,DIR,JTP0(IB),FINAL(IB),
1771 1 ' VAL=1,E12.4)
1772 C RUN ACROSS BOUNDARY AND SET CONDITIONS
1773 KMAX=IABS(MB1(IB)-MB2(IB))+IABS(NB1(IB)-NB2(IB))
1774 K=0
1775 DO 200 NB=1(IB),NB2(IB)
1776 DO 200 MB=1(IB),MB2(IB)
1777 K=K+1
1778 EL=RAMPT*FINAL(ISET1(IB))
1779 IF(KMAX.GT.0) EL=RAMPT*FLOAT(K-1)*FINAL(ISET2(IB))+
1780 1 FLOAT(KMAX+1-K)*FINAL(ISET1(IB))/FLOAT(KMAX)
1781 GOTO 120
1782 C 1. SIMPLE WATER LEVEL
1783 120 SEPP(N,M)=EL
1784 SEP(N,M)=.5*(SE(N,M)+SEPP(N,M))
1785 GOTO 190
1786 C 2. OUTWARD TRANSPORT
1787 130 IF(IABS(ITPO(IB)).EQ.1) THEN
1788 UHP(N,M+IUH)=F1*EL
1789 SEP(N,M)=SE(N,M+ISE)
1790 ELSE
1791 VHP(N+IUH,M)=F1*EL
1792 SEP(N,M)=SE(N+ISE,M)
1793 ENDIF
1794 SEP(N,M)=.5*(SE(N,M)+SEPP(N,M))
1795 GOTO 190
1796 C 3. RADIATION CONDITION, ORLANSKI
1797 140 IF(IABS(ITPO(IB)).EQ.2) GOTO 145
1798 DELSE=SE(N,M)-SE(N,M+ISE)
1799 IF(D(N,M+2*ISE).GT.0) DELSE=.5*SE(N,M)-2.*SE(N,M+ISE)
1800 1+.5*SE(N,M+2*ISE,M)
1801 SEP(N,M)=SE(N,M)-C1*SQRT(E+AG*.5*(D(N,M)+D(N,M+ISE)
1802 1.+SE(N,M)+SE(N,M+ISE))) *DELSE
1803 SEP(N,M)=.5*(SE(N,M)+SEPP(N,M))
1804 GOTO 130
1805 145 SE(N,M)-SE(N+ISE,M)
1806 IF(D(N+2*ISE,M).GT.0.) DELSE=1.5*SE(N,M)-2.*SE(N+ISE,M)
1807 1+.5*SE(N+2*ISE,M)
1808 SEP(N,M)=SE(N,M)-C1*SQRT(E+AG*.5*(D(N,M)+D(N+ISE,M)
1809 1.+SE(N,M)+SE(N+ISE,M))) *DELSE
1810 SEP(N,M)=.5*(SE(N,M)+SEPP(N,M))
1811 GOTO 190
1812 C 4. RADIATION CONDITION, RIEMANN INVARIANT
1813 150 IF(IABS(ITPO(IB)).EQ.2) GOTO 155
1814 SEP(N,M)=EL+F1*UH(N,M+IUH)/SQRT(E+AG*.5*(D(N,M)+D(N,M+ISE)
1815 1.+SE(N,M)+SE(N,M+ISE)))
1816 SEP(N,M)=.5*(SE(N,M)+SEPP(N,M))
1817 GOTO 190
1818 C Y-DIRECTION
1819 155 SE(N,M)=EL+F1*VH(N+IUH,M)/SQRT(E+AG*.5*(D(N,M)+D(N+ISE,M)
1820 1.+SE(N,M)+SE(N+ISE,M)))
1821 SEP(N,M)=.5*(SE(N,M)+SEPP(N,M))
1822 190 CONTINUE
1823 IF(ITEST.GT.0) WRITE(ISCR,210) N,EL,SEPP(N,M)
1824 210 FORMAT(3X,'N,M=',I2,I4,' EL=',F7.4,' SEPP=',F7.4)
1825 200 CONTINUE
1826 C RIVER FLOW BOUNDARIES
1827 220 IF(NUMRIV.EQ.0) RETURN
1828 IF(NSTI.EQ.1.AND.IPRNT1.GT.0) WRITE(ISCR,221) NSTI,NSTE,RAMPR
1829 221 FORMAT(1X,'BNDRY1: RIVER BOUNDARIES AT NSTI=',I6,' NSTE=',I6,
1830 1,' RAMPR',F4.2)
1831 CALL RR(YD,ISCR,LURIV,LENDRV,DRV,YRIV,NSIGR,NDRIV2,QRIV,QRATE)
1832 DO N=1,NSIGR
1833 RATE(N)=QRATE(ISETR(N))*RAMPR
1834 IF(NSTE.EQ.1.AND.IPRNT1.GT.0) WRITE(ISCR,225) N,ITPR(N),JTPR(N),
1835 1 ISETR(N),RATE(N)
1836 225 FORMAT(1X,'N=',I2,' IZ,',IZ,' ITPR,JTPR=',I2,I2,
1837 1 ' RATE X RAMPR (m**3/s)='F12.3)
1838 ENDDO
1839 DO 350 NR=1,NSIGR
1840 IS=0
1841 IF(ITPR(NR).LT.0) IS=-1
1842 IS=1+2*IS
1843 F1=FLOAT(ISS)/DL
1844 DO 350 MR1=NR1(NR),MR2(NR)
1845 DO 350 MR2=NR1(NR),NR2(NR)
1846 IF(JTPR(NR).LT.0) GOTO 330
1847 IF(JTPR(NR).GT.0) GOTO 330
1848 C WATER FALLS CONDITION
1849 SE(N,M)=SB(N,M)-DTE*RATE(NR)/(AREA(N,M)*DL*DL)
1850 F2=DTE*RATE(NR)/(AREA(N,M)*DL*DL)
1851 IF(NSTE.EQ.1.AND.IPRNT1.GT.0) WRITE(ISCR,*) RAMPR
1852 1 ' WATER FALLS AT N,M=',N,M,' DEL=',F2
1853 GOTO 350
1854 330 IF(IABS(ITPR(NR)).EQ.2) GOTO 340
1855 C INPUT FLOW, X-DIRECTION
1856 IBARR=MOD(IFIELD(N,M+IS),10)
1857 IF(IBM1.EQ.1.OR.IBARR.EQ.3) GOTO 350
1858 IF(NSTE.GT.1) UH(N,M+IS)=UHP(N,M+IS)
1859 UHP(N,M+IS)=F1*RATE(NR)/BX(N,M+IS)
1860 SEP(N,M)=SE(N,M+ISS)
1861 SE(N,M)=SE(N,M+ISS)
1862 GOTO 350
1863 C INPUT FLOW, Y-DIRECTION
1864 340 IBARR=MOD(IFIELD(N+IS,M),10)
1865 IF(IBM1.EQ.2.OR.IBARR.EQ.3) GOTO 350
1866 IF(NSTE.GT.1)VH(N+IS,M)=VHP(N+IS,M)
1867 VHP(N+IS,M)=F1*RATE(NR)/BY(N+IS,M)
1868 SEP(N,M)=SE(N+ISS,M)
1869 SE(N,M)=SE(N+ISS,M)
1870 350 CONTINUE
1871 360 RETURN
1872 END
1873 C -----
1874 C
1875 C
1876 SUBROUTINE HORVIS
1877 C APRIL 1986 K. HESS MEAD VAX-11/750 (REV 9/87)
1878 C PURPOSE - TO UPDATE THE HORIZONTAL EDDY VISCOSITY
1879 C VARIABLES -
1880 C AH,( ) = ARRAY TO STORE PRODUCT OF HORIZONTAL
1881 C VISCOSITY AND DEPTH, AT GRID CENTER
1882 C AH=AHO+CAH*DL*SQRT2(U,X)**2+2(V,Y)**2+(U,Y+V,X
1883 C AHO = BACKGROUND VISCOSITY
1884 C CAH = FACTOR FOR VISCOSITY
1885 INCLUDE 'COMM20.FOR'
1886 DIMENSION LS(LSIZE,NSIZE,MSIZE),VL(LSIZE,NSIZE,MSIZE)
1887 DATA ISMAG,ALFA/2,1,/
1888 C skip if necessary
1889 IF(IHVISC.LE.0.OR.MOD(NSTET,IHVISC).NE.0) GOTO 180
1890 C save 3-d velocity field
1891 DO M=1,MMAX
1892 DO N=1,NMAX
1893 DO L=1,LBOT
1894 UL(L,N,M)=UE(N,M)+ALFA*U(L,N,M)
1895 VL(L,N,M)=VE(N,M)+ALFA*V(L,N,M)
1896 ENDDO
1897 ENDDO
1898 ENDDO
1899 C LOOP OVER THE INTERIOR CELLS
1900 KOCN=10*(KOCNC-1)
1901 DO 120 M=1,MMAX
1902 NA=NAB(M)/1000
1903 NB=MOD(NAB(M),1000)
1904 IF(NA.GT.NB) GOTO 120
1905 MM=MIN0(M-1,MM)
1906 MP=MIN0(M,MM)
1907 DO 100 L=1,N,NE
1908 IF(IFIELD(N,M).LT.10.OR.IFIELD(N,M).GE.KOCN) GOTO 110
1909 NM=MAX0(N-1,1)
1910 NP=MIN0(N+1,NMAX)
1911 HD=(N,M)+SE(N,M)
1912 AH=0
1913 C INTERIOR CELLS
1914 DO 100 L=1,LBOT
1915 IF(INTER,EO,0) GOTO 90
1916 IF(ISMAG,EO,1) THEN
1917 DUUDY=.25*(UL(L,NP,M)+UL(L,N,M))*AMAX0(NFLUX(N,M),NFLUX(N,MP))
1918 1 -(UL(L,N,M)+UL(L,NM))*AMAX0(NFLUX(NM,M),NFLUX(NM,MP))
1919 2 +(UL(L,N,MP)+UL(L,N,M))*AMAX0(NFLUX(N,MM),NFLUX(N,M))
1920 3 -(UL(L,N,MM)+UL(L,N,M))*AMAX0(NFLUX(NM,MM),NFLUX(NM,MP))
1921 DVDX=.25*(VL(L,N,M)+VL(L,N,MP))*AMAX0(MFLUX(N,M),MFLUX(NP,M))
1922 1 -(VL(L,N,M)+VL(L,N,MP))*AMAX0(MFLUX(N,MM),MFLUX(NP,MM))
1923 2 +(VL(L,N,MP)+VL(L,N,M))*AMAX0(MFLUX(N,M),MFLUX(NM,M))
1924 3 -(VL(L,N,M)+VL(L,N,MM))*AMAX0(MFLUX(N,MM),MFLUX(NM,MM)))
1925 ELSE
1926 DUUDY=.25*(ABS(UL(L,NP,M)+UL(L,N,M))+ABS(UL(L,N,M)-UL(L,N,M))
1927 1 +ABS(UL(L,N,MP)+UL(L,N,M))-ABS(UL(L,N,MM)-UL(L,N,MP)))
1928 DVDX=.25*(ABS(VL(L,N,M)+VL(L,N,MP))+ABS(VL(L,N,M)-VL(L,N,MP))
1929 1 +ABS(VL(L,N,MP)+VL(L,N,M))-ABS(VL(L,N,MM)-VL(L,N,MP)))
1930 ENDIF
1931 ARG2=.25*((UL(L,N,M)-UL(L,N,MM))***2+
1932 1 ((UL(L,N,M)-UL(L,N,MM))***2)+(DUUDY+DVDX)*2
1933 C UPDATED VALUE
1934 90 AMM=AHO+CAH*DL*SORT(ARG)
1935 IF(AH3(L,N,M).EQ.0.0)AH3(L,N,M)=AMM*X
1936 IF(AH3(L,N,M).GT.0.0)AH3(L,N,M)=SQRT(AH3(L,N,M)*AMM*X)
1937 100 AH1=AH1+C1(L)*AH3(L,N,M)
1938 AH(N,M)=AH1
1939 110 CONTINUE
1940 120 CONTINUE
1941 C BOUNDARY CELLS
1942 IF(NUMOBC,LE,0) GOTO 140
1943 DO 130 IB=1,NUMOBC
1944 IDIR=IABS(ITPO(IB))
1945 IS=1*SIGN(1,ITPO(IB))
1946 DO 130 M=MB1(IB),MB2(IB)
1947 DO 130 N=NB1(IB),NB2(IB)
1948 HD=(N,M)+SE(N,M)
1949 C X-direction
1950 IF(IDIR,EO,1) THEN
1951 F1=H/(D(N,M+IS)+SE(N,M+IS))
1952 AH(N,M)=AH(N,M+IS)*F1
1953 DO 141 L=1,LBOT
1954 AH3(L,N,M)=AH3(L,N,M+IS)*F1
1955 ENDDO
1956 ENDIF
1957 C y-direction
1958 IF(IDIR,EO,2) THEN
1959 F2=H/(D(N+IS,M)+SE(N+IS,M))
1960 AH(N,M)=AH(N+IS,M)*F2
1961 DO 141 L=1,LBOT
1962 AH3(L,N,M)=AH3(L,N+IS,M)*F2
1963 ENDDO
1964 ENDIF
1965 130 CONTINUE
1966 C RIVER BOUNDARIES
1967 140 IF(NUMRIV,LE,0) GOTO 160
1968 DO 150 IB=1,NUMRIV
1969 IDIR=IABS(ITPR(IB))
1970 IS=1*SIGN(1,ITPR(IB))
1971 DO 150 M=MR1(IB),MR2(IB)
1972 DO 150 N=NR1(IB),NR2(IB)
1973 HD=(N,M)+SE(N,M)
1974 C z-direction
1975 IF(IDIR,EO,1) THEN
1976 F1=H/(D(N,M+IS)+SE(N,M+IS))
1977 AH(N,M)=AH(N,M+IS)*F1
1978 DO 141 L=1,LBOT
1979 AH3(L,N,M)=AH3(L,N,M+IS)*F1
1980 ENDDO
1981 ENDIF
1982 C y-direction
1983 IF(IDIR,EO,2) THEN
1984 F2=H/(D(N+IS,M)+SE(N+IS,M))
1985 AH(N,M)=AH(N+IS,M)*F2
1986 DO 141 L=1,LBOT
1987 AH3(L,N,M)=AH3(L,N+IS,M)*F2
1988 ENDDO

```

```

1989      ENDIF
1990 150  CONTINUE
1991 C     CORNER VALUES
1992 160  DO 170 M=1,MAX
1993      MP=MINO(M+1,MAX)
1994      DO 170 N=1,NMAX
1995      NP=MINO(N+1,NMAX)
1996      IF(IFIELD(N,M),LT,10)GOTO 170
1997      AHC(N,M)=.25*(AH(N,M)+AH(N,MP)+AH(NP,M)+AH(NP,MP))
1998 170  CONTINUE
1999 180  RETURN
2000      END
2001 C-----+
2002 C-----+
2003 C-----+
2004      SUBROUTINE UHVF
2005 C         JANUARY 1997 K.W. HESS
2006 C         PURPOSE - TO COMPUTE EXTERNAL-MODE FLOWRATES AND WATER LEVELS
2007 C         FROM REVISED EQUATIONS FOR VARIABLE GRID WIDTH (TO
2008 C         MODEL NARROW RIVER WIDTHS).
2009 C-----+
2010 C-----+
2011 C-----+
2012 C-----+
2013 C-----+
2014 C-----+
2015      INCLUDE 'COM20.FOR'
2016      BETAH=MAX0(0,MIN0(1,IBETAH))
2017      BETAA=MAX0(0,MIN0(1,IBETAA))
2018      CX1=.25*DTE/DL
2019      CX2=.5*AG*DTE/DL
2020      CX3=DTE/DL
2021      CX4=DTE/DL**2
2022      CX5=.5*DTE/DL
2023      CX6=.25*AG*DTE/DL
2024      CX7=1./CX1
2025      CX8=1./CX5
2026      CX9=2.*DTE/DL**2 ! additional constant
2027      CX10=0.1/RHOW
2028      GAMMA=1.0
2029      IF(IBOTV.EQ.0)GAMMA=.0
2030      IF(IBOTV.EQ.3)GAMMA=.5
2031 C-----+
2032      X-DIRECTION SWEEP
2033 C-----+
2034      DO 220 NC=1,NCOL
2035      N=ICOL(1,NC)
2036      MA=ICOL(2,NC)
2037      MB=ICOL(3,NC)
2038      ILF=ICOL(4,NC)
2039      IRGT=ICOL(5,NC)
2040      MAC=MA-1
2041      NM=MAX0(N-1,1)
2042      NP=MINO(N+1,NMAX)
2043      SWEEP DOWN THE COLUMN
2044      DO 200 M=MM,MB
2045      MM=MAX0(M-1,1)
2046      MP=MINO(M+1,MAX)
2047      MP=MINO(M+2,MMAX)
2048      HS=D(N,M)+BETAH*SE(N,M)+E
2049      HP=D(N,MP)+BETAH*SE(N,MP)+E
2050      HM=D(N,MM)+BETAH*SE(N,MM)+E
2051      HPP=D(N,MPP)+BETAH*SE(N,MPP)+E
2052      HB=.5*(HS+HP)
2053      AREAP=AREA(N,M)
2054      HH=CX1*(HB+CX5)
2055      C TOTAL STRESS FORMULATION
2056      FBAR=.5*(PHI(N,M)+PHI(N,MP))
2057      EDGE=AMIN1(FEDGE(N,M),FEDGE(N,MP))
2058      XDT=DTE*(EDGE*.5*(TSX(N,M)+TSX(N,MP))-CX10*HB*DPADX-HB*)
2059      1 GSTARX(N,M)-FBAR*(GAMMA*X(LBOT,N,M)*(1.-GAMMA)*X(LBOT-1,N,M))
2060      CHANNEL: NO: BETAC=0 YES: BETAC=1
2061      BETAC=1-IFIX(BX(N,M))
2062      DTFNM=DTE*(FBAR/HB+BETAC*THETSU(N,M))
2063      LOWER END BOUNDARY CONDITIONS
2064      IF(M.GT.MM)GOTO 160
2065      AREAP=AREA(N,MP)
2066      DMP=AREAP*(N,MP)-CX1*(BX(N,MP)*UH(N,MP)-EX(N,M)*UH(N,M)
2067      1 +2.*(BY(N,MP)*VH(N,M)-BY(NM,MP)*VH(NM,MP)))
2068      C WATER LEVEL CONDITION
2069      IF(ILFT.GT.0)GOTO 150
2070      DENOM=1./(1.+DTFN)
2071      FA(M)=(UH(N,M)+XDT+HH*SEP(N,M))*DENOM
2072      FB(M)=HH*DENOM
2073      GOTO 200
2074 C-----+
2075      FLOWRATE CONDITION
2076 150  FA(M)=UH(N,M)
2077      FB(M)=0.0
2078      GOTO 200
2079 C-----+
2080      INTERIOR COMPUTATIONAL GRIDS: MASS RECURSION
2081 160  AREAX=AREAP
2082      DENOM=1./((AREAX-CX1*EX(N,MM)-FB(MM))*DENOM
2083      GA(M)=(DCP*CX1*BX(N,MM)*FA(MM))*DENOM
2084      GB(M)=CX1*BX(N,M)*DENOM
2085      CHECK FOR UPPER BOUNDARY (M=MB)
2086      IF(M.EQ.MB)GOTO 180
2087      START COMPUTING THE MOMENTUM RECURSION ARRAYS
2088      VB=(VH(N,M)+VH(N,MP)+VH(NM,MM))+HVN(M,MM)) / (FLOAT(NFLUX(N,M)+
2089      1 NFLUX(N,MP)+NFLUX(NM,MM)+NFLUX(NM,MP))+E)
2090      FCOR=EDGE*(FCOR0+DFDM*FLOAT(M-MCOR)+DFDN*FLOAT(N-NCOR))
2091      AREAP=AREA(N,MP)
2092      DMP=AREAP*(N,MP)-CX1*(BX(N,MP)*UH(N,MP)-EX(N,M)*UH(N,M)
2093      1 +2.*(BY(N,MP)*VH(N,M)-BY(NM,MP)*VH(NM,MP))
2094      BXI=1.0/BX(N,M)
2095      BP=CX9*(1.-BX(N,MM)*BXI)*AH(N,MP)
2096      EM=CX9*(1.-BX(N,MM)*BXI)*AH(N,M)
2097      NON-LINEAR TERMS
2098      AUU=BETAA*CX3*(THETA3(N,M)-THETA3(N,MM))
2099      CU=SIGN(1.0,UH(N,M))
2100      FN1=BETAA*CX5*(1.-CU)*EX1*THETA1(N,MM)
2101      FN2=BETAA*CX5*(1.-CU)*(1.+CU)*BX1*THETA1(N,M)
2102      FN3=BETAA*CX5*(1.-CU)*EX1*THETA1(N,MM)
2103      AHDUYY=CX4*(AHC(N,M)*(UE(NP,M)-UE(N,M)+VE(N,MP)-VE(N,M))-1.
2104      AHC(NM,M)*(UE(N,M)-UE(NM,M)+VE(NM,MP)-VE(NM,MM)))*(1.-BETAC)
2105      F1=-BM/(HS+HM)-FN1
2106      F2=1.+DTFN*(BP+BM)/(HS+HP)+FN2
2107      F3=-BP/(HP+HPB)+FN3
2108      F4=UH(N,M)+XDT+AHDUYY+FCOR*VB-AUV
2109      BXIP=F3/BX(N,MP)
2110      DENOM=1./(GB(M)*(HH+F1*FB(MM))+F2+BX(N,M)*BXIP)
2111      FA(M)=(F4-F1*FA(MM)+GA(M)*(F1*FB(MM)+HH)-DMP*CX7*BXIP)*DENOM
2112      FB(M)=(HH-AREAP*CX7*BXIP)*DENOM
2113      GOTO 200
2114 C-----+
2115      UPPER END BOUNDARY CONDITIONS
2116      CONTINUE
2117      DO 180 M=MA,MB
2118      UH(N,M)=UH(N,M)+XDT+HH*GA(M)*DENOM
2119      F1=UH(N,M)+XDT+HH*GA(M)
2120      GOTO 200
2121 190  FA(M)=UH(N,M)
2122      FB(M)=0.0
2123 200  CONTINUE
2124 C-----+
2125      LOOP BACK DOWN THE COLUMN, CALCULATING FLOWRATE & WATER LEVEL
2126      DO 210 MM=MA,MB
2127      M=MA+MB-MM
2128      UH(N,M)=FA(M)-FB(M)*SEP(N,M+1)
2129      SEP(N,M)=GA(M)-GB(M)*UH(N,M)
2130      MAKE SURE DEPTH IS POSITIVE
2131      IF(SEP(N,M)+D(N,M).GT.0.0)GOTO 210
2132      NEGS=NEGS+1
2133 210  CONTINUE
2134      UH(N,M)=FA(MAM)-FB(MAM)*SEP(N,M)
2135 220  CONTINUE
2136 C-----+
2137      Y-DIRECTION SWEEP
2138 C-----+
2139 280  DO 370 NR=1,NROW
2140      M=IROW(1,NR)
2141      NA=IROW(2,NR)
2142      NB=IROW(3,NR)
2143      ILFT=IROW(4,NR)
2144      IRGT=IROW(5,NR)
2145      NAM=NA-1
2146      MM=MAX0(M-1,1)
2147      MP=MINO(M+1,MMAX)
2148 C-----+
2149      SWEEP ACROSS ROWS
2150      DO 350 N=NM,NB
2151      NM=MAX0(N-1,1)
2152      NP=MINO(N+1,NMAX)
2153      HS=D(N,M)+BETAH*SE(N,M)+E
2154      HP=D(N,MP)+BETAH*SE(N,MP)+E
2155      HM=D(N,MM)+BETAH*SE(N,MM)+E
2156      HPP=D(N,MPP)+BETAH*SE(N,MPP)+E
2157      HB=.5*(HS+HP)
2158      AREAP=AREA(N,M)
2159      HH=CX6*(HB+HP)
2160 C-----+
2161      EXTERNAL FORCES
2162      FBAR=.5*(PHI(N,M)+PHI(N,MP))
2163      EDGE=AMIN1(FEDGE(N,M),FEDGE(N,MP))
2164      YDT=DTE*(EDGE*.5*(TSY(N,M)+TSY(N,MP))-CX10*HB*DPADY-HB*)
2165 C-----+
2166 1 GSTAR(N,M)-FBAR*(GAMMA*X(LBOT,N,M)+(1.-GAMMA)*X(V(LBOT-1,N,M)))
2167      CHANNEL: NO:BETAC=0 YES:BETAC=1
2168      BETAC=1-IFIX(BY(N,M))
2169      DTFNM=DTE*(FBAR/HB+BETAC*THETSV(N,M))
2170      CHECK FOR LOWER END BOUNDARY CONDITIONS
2171      IF(N.GT.NAM)GOTO 310
2172      AREAP=AREA(NP,M)
2173      DNPN=AREAP*SEP(NP,M)-CX1*(BX(NP,M)*(UH(NP,M)+UH(NP,MM)))
2174      1 -BX(NP,MM)*(UH(NP,MM)+UH(NP,MM))
2175      WATER LEVEL CONDITION
2176      IF(ILFT.GT.0)GOTO 300
2177      DENOM=1./(1.+DTFN)
2178      FA(N)=(VH(N,M)+YDT+HH*(SEPP(N,M)-SE(NP,M)+SE(N,M)))*DENOM
2179 C-----+
2180      FLOWRATE CONDITION
2181      FA(N)=UH(N,M)
2182      FB(N)=0.0
2183 C-----+
2184 310  INTERIOR COMPUTATIONAL GRIDS: MASS RECURSION
2185      CHANNEL: NO:AREAP=AREAP
2186      DENOM=1./((AREAX+CX5*BY(NM,M)*FB(NM))
2187      GA(N)=(DNP+CX5*BY(NM,M)*FA(NM))*DENOM
2188      GB(N)=CX5*BY(N,M)*DENOM
2189      CHECK FOR UPPER END (N=NB)
2190 C-----+
2191      IF(N.EQ.NB) GOTO 330
2192      START COMPUTING THE MOMENTUM RECURSION ARRAYS
2193      UB=(UH(N,M)+UH(N,MP)+UH(N,MM)+UH(NP,MM)) / (FLOAT(MFLUX(N,M)+
2194      1 MFLUX(N,MP)+MFLUX(N,MM)+MFLUX(NP,MM))+E)
2195      FCOR=EDGE*(FCOR0+DFDM*FLOAT(M-MCOR)+DFDN*FLOAT(N-NCOR))
2196      AREAP=AREA(NP,M)
2197      DNPN=AREAP*SEP(NP,M)-CX1*(BX(NP,M)*(UH(NP,M)+UH(NP,MM)))
2198      BY1=1.0/BY(N,M)
2199      BP=CX9*(1.+BY(NM,M)*BY1)*AH(N,M)
2200 C-----+
2201      BM=CX9*(1.+BY(NM,M)*BY1)*AH(N,M)
2202      NON-LINEAR TERMS
2203      AUU=BETAA*CX3*(THETA3(N,M)-THETA3(N,MM))
2204      CU=SIGN(1.0,UH(N,M))
2205      FN1=BETAA*CX5*(1.+CU)*BY1*THETA2(N,M)
2206      FN2=BETAA*CX5*(1.-CU)*BY1*THETA2(N,M)
2207      FN3=BETAA*CX5*(1.-CU)*BY1*THETA2(N,M)
2208      LATERAL VISCOSITY TERM
2209      AHDXVX-CX4*(AHC(N,M)*(UE(NP,M)-UE(N,M)+VE(N,MP)-VE(N,M))-1.
2210      AHC(N,M)*(UE(NP,MM)-UE(N,MM)+VE(NM,MM)+VE(N,M)-VE(N,MM)))*(1.-BETAC)
2211      SET RECURSION ARRAY HERE
2212      F1=-BM/(HS+HM)-FN1
2213      F2=1.+DTFN*(BP+BM)/(HS+HP)+FN2
2214      F3=-BP/(HP+HPB)+FN3
2215      F4=VH(N,M)+YDT+AHDXVX-FCOR*UB-AUV-HH*(SE(NP,M)-SE(N,M))
2216      BY1P=F3/BY(N,P)
2217      DENOM=1./(GB(N)*(HH+F1*FB(NM))+F2+BY1P*BY(N,M))
2218      FA(N)=(F4-F1*FA(NM)+GA(N)*(HH+F1*FB(NM))-DNP*CX8*BY1P)*DENOM
2219      FB(N)=(HH-CX8*AREAP*BY1P)*DENOM
2220      GOTO 350
2221      C-----+
2222      UPPER END BOUNDARY CONDITIONS
2223 330  IF(ILFT.GT.0)GOTO 340
2224      DENOM=1./(1.+HH*GB(N)+DTFN)
2225      FA(N)=(VH(N,M)+YDT+HH*(GA(N)-SE(NP,M)+SE(N,M)))*DENOM
2226      FB(N)=0.0
2227 350  CONTINUE
2228 C-----+
2229      LOOP BACK DOWN THE ROW, CALCULATING FLOWRATE AND WATER LEVEL
2230      DO 360 NN=NA,NB
2231      N=NA+NB-NN
2232      VHP(N,M)=FA(N)-FB(N)*SEPP(N+1,M)
2233      SEP(N,M)=GA(N)-GB(N)*VHP(N,M)
2234      MAKE SURE DEPTHS ARE POSITIVE
2235      IF(SEP(N,M)+D(N,M).GT.0.0)GOTO 360
2236      NEGS=NEGS+1
2237 360  CONTINUE
2238      VHP(N,M)=FA(NAM)-FB(NAM)*SEPP(NA,M)
2239 370  CONTINUE
2240 C-----+

```

```

2241 C      UPDATE AND SAVE VARIABLES
2242 F1=2.
2243 F2=1.
2244 IF(NSTE.EQ.ISPLIT)THEN
2245   F1=1.
2246   F2=.5/FLOAT(ISPLIT)
2247 END IF
2248 DO 390 N=1,NMAX
2249 DO 390 M=1,MMAX
2250 UH(N,M)=UHF(N,M)
2251 VH(N,M)=VHF(N,M)
2252 SE(N,M)=SEPP(N,M)
2253 UHOLD(N,M)=F2*(UHOLD(N,M)+F1*UH(N,M))
2254 VHOLD(N,M)=F2*(VHOLD(N,M)+F1*VH(N,M))
2255 C      COMPUTE EXTERNAL-MODE VELOCITY
2256 DO 400 N=1,NMAX
2257 NP=MINO(N+1,NMAX)
2258 DO 400 M=1,MMAX
2259 MP=MINO(M+1,MMAX)
2260 UE(N,M)=UH(N,M)*(1.+FLOAT(MFLUX(N,M)))/
2261 1*((D(N,M)+D(N,MP))+BETAH*(SE(N,M)+SE(N,MP)))+E)
2262 VE(N,M)=VH(N,M)*(1.+FLOAT(NFLUX(N,M)))/
2263 1*((D(N,M)+D(N,MP))+BETAH*(SE(N,M)+SE(N,MP)))+E)
2264 400 CONTINUE
2265 C
2266 RETURN
2267 END
2268 C=====
2269 C      FILE INTRNL.FOR
2270 C-----
2271 C      SUBROUTINE INTRNL
2272 C      MAY 1988 K. W. HESS TDL
2273 C      PURPOSE - TO COMPUTE THE INTERNAL MODE VARIABLES WITH VARIABLE
2274 C      WIDTH. THESE ARE THE VERTICAL VELOCITIES, EDDY VISCO
2275 C      AND HORIZONTAL VELOCITY DEPARTURES FROM THE VERTICAL
2276 C      MEAN.
2277 C      VARIABLES -
2278 C      ISKIP = INDEX FOR SKIPPING THE UPDATE OF THE EDDY VISO
2279 C
2280 C
2281 INCLUDE 'COMM20.FOR'
2282 C      CHECK FOR SKIPPING ALL INTERNAL-MODE CALCS
2283 IF(INTER.EQ.0)GOTO 100
2284 C      GET INTERNAL-MODE BOUNDARY CONDITIONS FOR NON-WATERFALLS
2285 CALL BNDRY2
2286 C      UPDATE TURBULENT MOMENTUM TRANSFER COEFFICIENTS.
2287 IF(IVISC.GT.0.AND.(NSTI.LE.IHR.OR.MOD(NSTI,IVISC).EQ.0))GOTO 100
2288 C      CALL VERRIS
2289 C      GET INTERNAL-MODE VELOCITIES
2290 CALL UPVE
2291 C      GET VERTICAL VELOCITY
2292 CALL WVERT
2293 C      GET UPDATED THETA'S
2294 100 CONTINUE
2295 RETURN
2296 END
2297 C
2298 C-----
2299 C      SUBROUTINE BNDRY2
2300 C      OCTOBER 1984 K. W. HESS MEAD VAX11/750
2301 C      PURPOSE - TO SET THE RIVER'S INTERNAL-MODE VELOCITY BOUNDARY
2302 C      CONDITIONS FOR NON-FALLS CONDITION.
2303 C      VARIABLES
2304 C      QRIV = RIVER FLOWRATE (M**3/S)
2305 C      JTPR() = RIVERINE CONDITION: 1=FLUME, 2=WATER FALLS
2306 C      INCLUDE 'COMM20.FOR'
2307 C      RIVER FLOW BOUNDARIES
2308 C      IF(NUMRIV.LE.0)GOTO 130
2309 C      LOOP THRU THE RIVERS
2310 C      DO 120 NR=1,NUMRIV
2311 C      IF(JTPR(NR).EQ.1)GOTO 120
2312 C      FLUME CONDITIONS. SET DIRECTION AND SENSE
2313 C      IDIR=IABS(JTPR(NR))
2314 C      IS=0
2315 C      IF(JTPR(NR).LT.0)IS=-1
2316 C      F1=1.
2317 C      IF(JTPR(NR).LE.0)F1=-1.
2318 C      F1=ISIGN(1,JTPR(NR))
2319 C      IS=(1-ISIGN(1,JTPR(NR)))/2
2320 C      LOOP THRU CELLS AT BOUNDARY
2321 C      UTOP=0.
2322 C      DO 110 M=NR1(NR),MR2(NR)
2323 C      DO 110 N=NR1(NR),NR2(NR)
2324 C      SET NEW VELOCITIES
2325 C      F2=F1*UTOP
2326 C      IBARR=0
2327 C      IF(IDIR.EQ.1)IBARR=MOD(IFIELD(N,M+IS),10)
2328 C      IF(IDIR.EQ.2)IBARR=MOD(IFIELD(N+IS,M),10)
2329 C      IF(IDIR.EQ.1.AND.(IBARR.EQ.1.OR.IBARR.EQ.3))F2=0.0
2330 C      IF(IDIR.EQ.2.AND.IBARR.GE.2)F2=0.0
2331 C      F3=PI*DQ
2332 C      DO 100 L=1,LBOT
2333 C      FVE=COS(F3*FLOAT(L-1))
2334 C      IF(IDIR.EQ.1)V(L,N,M+IS)=F2*FVE
2335 C      IF(IDIR.EQ.2)V(L,N+IS,M)=F2*FVE
2336 C      100 CONTINUE
2337 110 CONTINUE
2338 120 CONTINUE
2339 130 RETURN
2340 C
2341 END
2342 C
2343 C-----
2344 C      FUNCTION FRHO(SAL,TMP)
2345 C      JUNE 1996 K. W. HESS CEOB SGI
2346 C      PURPOSE - TO GENERATE THE ADDED WATER DENSITY DUE TO SALINITY
2347 C      (S) AND TEMPERATURE (T) DEG. C. THAT IS,
2348 C      RHO = RHOW + 1000.*FRHO(S,T) kg/m**3
2349 C      MANAEV FORMULATION
2350 C      SS=AMAX1(SAL,0.0)
2351 C      TT=AMAX1(TMP,0.0)
2352 C      FRHO= 7.E-5+SS*(8.02E-4-2.0E-6*TT)-TT*(3.5E-6+4.69E-6*TT)
2353 C      RETURN
2354 C
2355 END
2356 C
2357 C-----
2358 C      FUNCTION FRHO2(S1,T1)
2359 C      UNESCO FORMULATION
2360 C      DATA RHOO/1000./
2361 C      TR=AMAX1(0.0,T1)
2362 C      SR=AMAX1(0.0,S1)
2363 C      RHOR = 999.842594 + 6.793952E-2*TR
2364 C      $      - 9.095290E-3*TR**2 + 1.001685E-4*TR**3
2365 C      $      - 1.120083E-6*TR**4 + 6.536332E-9*TR**5
2366 C
2367 RHOR = RHOR + (0.824493 - 4.0899E-3*TR
2368 $      + 7.6438E-5*TR**2 - 8.2467E-7*TR**3
2369 $      + 5.3875E-9*TR**4) * SR
2370 $      + (-5.72466E-3 + 1.0227E-4*TR
2371 $      - 1.6546E-6*TR**2) * SR**1.5
2372 $      + 4.8314E-4 * SR**2
2373 FRHO=(RHOR-RHO)*1.E-3
2374 RETURN
2375 END
2376 C -----
2377 C
2378 C      SUBROUTINE GRADP(N,M,INTGRL)
2379 C      PURPOSE - TO COMPUTE HORIZONTAL GRADIENTS DUE TO DENSITY.
2380 C      NOTE: RHO = RHOW + FRHO(S,T)
2381 C      VARIABLES -
2382 C      INTGRL = INDEX TO UPDATE INTEGRATED PRESSURE:l=YES
2383 C      INCLUDE 'COMM20.FOR'
2384 C      SET INITIAL GRADIENTS
2385 C
2386 DO 100 L=1,LBOT
2387 GRX(L)=0.
2388 100 GRY(L)=0.
2389 IF((ICOUPLE.LE.1.OR.IBETAP.EQ.0.OR.RAMPG.EQ.0.)RETURN
2390 C      SET CONSTANTS
2391 MP=MINO(M+1,MMAX)
2392 NP=MINO(N+1,NMAX)
2393 C01=RAMPG*AG/DL
2394 C02=C01*DO
2395 C      SET BATHYMETRY VALUES
2396 H1=(D(N,M)+SE(N,M))
2397 H2=(D(N,MP)+SE(N,MP))
2398 DELSX=SE(N,MP)-SE(N,M)
2399 DELHX=H2-H1
2400 H3=(D(N,MP)+SE(NP,M))
2401 DELSY=SE(NP,M)-SE(N,M)
2402 DELHY=H3-H1
2403 C      INITIALIZE PRESSURE AND DENSITY TERMS
2404 SX=0.
2405 SXP=0.
2406 SY=0.
2407 SYP=0.
2408 PX=0.
2409 PY=0.
2410 C      LOOP OVER DEPTHS
2411 IF(MFLUX(N,M).EQ.0.OR.H2.LE.0.0)GOTO 120
2412 C      X-DIRECTION GRADIENTS
2413 DO 110 L=1,LBOT
2414 Q=FLOAT(L-1)*DO
2415 SAVG=.5*(S(L,N,M)+S(L,N,MP))
2416 TAVG=.5*(T(L,N,M)+T(L,N,MP))
2417 GRX(L)=(DELSX+DELHX*Q)*C01*FRHO(SAVG,TAVG)
2418 IF(L.EQ.1)GOTO 105
2419 SAVGM=.5*(S(L,N,MP)+S(L-1,N,MP))
2420 TAVGM=.5*(T(L,N,MP)+T(L-1,N,MP))
2421 SXP=SXP+C01*FRHO(SAVGM,TAVGM)
2422 SAVGM=.5*(S(L,N,M)+S(L-1,N,M))
2423 TAVGM=.5*(T(L,N,M)+T(L-1,N,M))
2424 SX=SX+C02*FRHO(SAVGM,TAVGM)
2425 GRX(L)=GRX(L)+(SXP*H2-SX*H1)
2426 105 PX=PX+C1(L)*GRX(L)
2427 110 CONTINUE
2428 C      Y-DIRECTION
2429 120 IF(NFLUX(N,M).EQ.0.OR.H3.LE.0.0)GOTO 140
2430 DO 130 L=1,LBOT
2431 Q=FLOAT(1-L)*DO
2432 SAVG=.5*(S(L,N,M)+S(L,N,MP))
2433 TAVG=.5*(T(L,N,M)+T(L,N,MP))
2434 GRY(L)=(DELY+DELHY*Q)*C01*FRHO(SAVG,TAVG)
2435 C      IF(L.EQ.1)GOTO 130
2436 IF(L.EQ.1)GOTO 125
2437 SAVGN=.5*(S(L,N,M)+S(L-1,N,MP))
2438 TAVGN=.5*(T(L,N,M)+T(L-1,N,MP))
2439 SYP=SYP+C01*FRHO(SAVGN,TAVGN)
2440 SAVGN=.5*(S(L,N,M)+S(L-1,N,M))
2441 TAVGN=.5*(T(L,N,M)+T(L-1,N,M))
2442 SY=SY+C02*FRHO(SAVGN,TAVGN)
2443 GRY(L)=GRY(L)+(SY*H3-SY*H1)
2444 125 PY=PY+C1(L)*GRY(L)
2445 130 CONTINUE
2446 140 CONTINUE
2447 C      INTEGRATED PRESSURE
2448 IF(INTGRL.NE.1)GOTO 150
2449 GSTARX(N,M)=PX
2450 GSTARY(N,M)=PY
2451 150 RETURN
2452 END
2453 C
2454 C-----
2455 C      SUBROUTINE THETAS
2456 C      JAN 1997 K.W.HESS
2457 C      PURPOSE - TO COMPUTE THE THETA FUNCTIONS.
2458 C      VARIABLES -
2459 C      THETA1(,) = BX*UE*INTEGRAL OVER DEPTH OF:(1 + U/UE)**2
2460 C      THETA2(,) = BY*VE*INTEGRAL OVER DEPTH OF:(1 + V/VE)**2
2461 C      THETA3(,) = UE*VE*INTEGRAL OVER DEPTH OF:(1+U/UE)*(1+V/VE)
2462 C      THETUS(,) = INTEGRAL OVER DEPTH OF (1 + U/UE)*ABS(1 + U/UE)
2463 C      TIMES CDRGWS/H/BX
2464 C      THETVS(,) = INTEGRAL OVER DEPTH OF (1 + V/VE)*ABS(1 + V/VE)
2465 C      TIMES CDRGWS/H/BY
2466 C
2467 INCLUDE 'COMM20.FOR'
2468 MMAX=MMAX-1
2469 NMMAX=NMAX-1
2470 DO 250 M=1,MMAX
2471 DO 250 N=1,NMAX
2472 IF(IFIELD(N,M).LT.10)GOTO 250
2473 C      EXTERNAL-MODE ONLY
2474 IF(INTER.NE.0)GOTO 150
2475 TH1=1.0
2476 TH2=1.0
2477 TH3=.25*(UE(N,M)+UE(N+1,M)) * (VE(N,M)+VE(N,M+1))
2478 TS1=1.0
2479 TS2=1.0
2480 GOTO 210
2481 C      INTERNAL MODE. GET RECIPROCAL OF UE, VE
2482 150 UBI=1./SIGN(AMAX1(ABS(UE(N,M)),0.0001),UE(N,M))
2483 VBI=1./SIGN(AMAX1(ABS(VE(N,M)),0.0001),VE(N,M))
2484 C      SUM OVER DEPTH
2485 TH1=0.0
2486 TH2=0.0
2487 TH3=0.0
2488 TS1=0.0
2489 TS2=0.0
2490 DO 200 L=1,LBOT
2491 IF(IBETAA.LE.0)GOTO 180
2492 TH1=TH1+C1(L)*(1.+U(L,N,M)*UBI)**2

```

```

2493 TH2=TH2+CI(L)*(1.+V(L,N,M)*VBI)**2
2494 TH3=TH3+CI(L)*.25*(UE(N,M)+U(L,N,M)+UE(N+1,M)+U(L,N+1,M))* 
2495 1 *(VE(N,M)+V(L,N,M)+VE(N,M+1)+V(L,N,M+1))
2496 180 IF(BX(N,M),LT,1.,0) TS1=TS1+CI(L)*(1.+U(L,N,M)*UBI)*ABS(1.+
2497 1 U(L,N,M)*UBI)
2498 IF(BY(N,M),LT,1.0) TS2=TS2+CI(L)*(1.+V(L,N,M)*VBI)*ABS(1.+
2499 1 V(L,N,M)*VBI)
2500 200 CONTINUE
2501 C CHECK MAGNITUDES OF NON-LINEAR TERM INTEGRALS
2502 210 IF(IBETAA.LE.0) GOTO 230
2503 THETA1(N,M)=.5*(THETA1(N,M)+AMINI(10.0,TH1)*UE(N,M)*BX(N,M))
2504 THETA2(N,M)=.5*(THETA2(N,M)+AMINI(10.0,TH2)*VE(N,M)*BY(N,M))
2505 TH3=SIGN(AMINI(ABS(TH3),10.,0),TH3)
2506 IF(MOD(IFIELD(N,M),10).GT.0.OR.IFIELD(N+1,M).LT.10) TH3=0.0
2507 1 IFIELD(N,M+1),LT,10.OR.IFIELD(N+1,M+1).LT.10) TH3=0.0
2508 THETA3(N,M)=.5*(THETA3(N,M)+TH3)
2509 C SIDE FRICTION TERMS
2510 230 THETSU(N,M)=CDRGWS*TS1*ABS(UE(N,M))/(DL*BX(N,M))
2511 THETSV(N,M)=CDRGWS*TS2*ABS(VE(N,M))/(DL*BY(N,M))
2512 250 CONTINUE
2513 RETURN
2514 END
2515 C -----
2516 C -----
2517 C -----
2518 SUBROUTINE GETCJ(CJ,CI,LBOT,LSIZE,IBOTV)
2519 C FACTOR TO REDISTRIBUTE ANY NON-ZERO INTERNAL-MODEL VELOCITY,
2520 C MAINTAINING SAME BOTTOM VELOCITY IF IBOTV=0.
2521 DIMENSION CJ(LSIZE),CI(LSIZE)
2522 SUM=0.
2523 DO L=L,BOT
2524 CJ(L)=1.
2525 IF(IBOTV.EQ.0) CJ(L)=MAX0(MINO(LBOT-L,2),0)
2526 SUM=SUM+CJ(L)*CJ(L)
2527 ENDO
2528 DO L=L,BOT
2529 CJ(L)=CJ(L)/SUM
2530 ENDO
2531 RETURN
2532 END
2533 C -----
2534 C -----
2535 C -----
2536 SUBROUTINE UPVP
2537 C
2538 C MARCH 1996 K. W. HESS CEOB
2539 C PURPOSE - TO COMPUTE THE INTERNAL MODE VARIABLES WITH VARIABLE
2540 C WIDTH. THESE ARE THE VERTICAL VELOCITIES, EDDY VISCONS
2541 C AND HORIZONTAL VELOCITY DEPARTURES FROM THE VERTICAL
2542 C (SEE MECCA PROGRAM DOCUMENTATION, PART B)
2543 C INCLUDES NON-LINEAR TERMS W/O IF STATEMENTS AND
2544 C HAS 3-D HORIZONTAL VISCOSEITY.
2545 C
2546 C VARIABLES -
2547 C IBOTV = BOTTOM B.C. INDEX:
2548 C 0 : U=0
2549 C 1 : AvDU/DZ = TBX first order DU/DZ
2550 C 2 : AvDU/DZ = TBX second order DU/DZ
2551 C 3 : AvDU/DZ = TBX log-layer, mid-level
2552 C ITOPV = ORDER OF TOP B.C. DERIVATIVE
2553 C 1 : FIRST ORDER DU/DZ
2554 C 2 : SECOND.
2555 C 3 : FIRST-ORDER TOTAL
2556 C
2557 INCLUDE 'COMM20.FOR'
2558 DIMENSION UP(LSIZE),UPM(LSIZE,NSIZE),UPMM(LSIZE,NSIZE),VP(LSIZE),
2559 2 VPMM(LSIZE,NSIZE),VPMM(LSIZE,NSIZE),FBC(LSIZE),CJ(LSIZE)
2560 C NONLINEAR TERMS: INCLUDE IF IBETAA=1
2561 FNOLN=0.
2562 IF(IBETAA.NE.0) FNOLN=1.
2563 C NONLINEAR TERMS:FOR NONLN, 1=NO, 2=YES
2564 BETAH=MAX0(0,MINO(1,IBETAH))
2565 KOCN=10.*KOCNBC
2566 KRIV=10.*(KOCNBC+1)
2567 LAYRM=LAYRS-1
2568 DIFF=0.0
2569 C
2570 B1=DTI/(8.*DQ)
2571 B2=DTI/(2.*DQ**2)
2572 B3=DTI/(2.*DQ**2)
2573 B4=DTI/DI**2
2574 B5=DTI/(2.*DL)
2575 B6=DTI/(2.*DL)
2576 B7=DTI/(16.*DL)
2577 B8=DTI/(4.*DL)
2578 B9=2./3.
2579 B10=4./3.
2580 FSPLIT=ISPLIT
2581 C
2582 DO 90 L=1,LBOT
2583 FBC(L)=0.0
2584 IF(ITOPV.EQ.2) FBC(2)=0.33333
2585 C
2586 C BEGIN LOOP THRU THE MESH
2587 DO 550 M=1,MMAX
2588 C STORE THE PRESENT VELOCITIES
2589 DO 120 N=1,NMAX
2590 DO 120 L=1,LBOT
2591 IF(M.GT.1)GOTO 100
2592 UPMM(L,N)=0.0
2593 VPMM(L,N)=0.0
2594 GOTO 110
2595 100 UPMM(L,N)=UPM(L,N)
2596 IF(M.EQ.1)UPMM(L,N)=2.*UPM(L,M)-U(L,N,M+1)
2597 VPMM(L,N)=VP(L,N)
2598 110 UP(L,N)=U(L,N,M)
2599 VP(L,N)=V(L,N,M)
2600 IF(N.EQ.1)VPM(L,M)=2.*V(L,N,m)-V(L,N+1,M)
2601 120 CONTINUE
2602 C DEFINE THE COLUMN LIMITS
2603 NA=NAB(M)/1000
2604 NB=NAB(M)/1000*NA
2605 IF(NA.GT.NB)GOTO 550
2606 DO 545 N=NA,NB
2607 II=IFIELD(N,M)
2608 IF(II,LT,10) GOTO 540
2609 IBARR=MOD(II,10)
2610 IF(IBARR.EQ.3) GOTO 540
2611 C GET THE INTERNAL PRESSURE GRADIENT
2612 CALL GRADP(N,M,1)
2613 MM=MAX0(N-1,1)
2614 MP=MIN0(N+1,MMAX)
2615 NM=MAX0(N-1,1)
2616 NP=MIN0(N+1,NMAX)
2617 C X-DIRECTION HORIZONTAL VELOCITY DEPARTURE
2618 C
2619 C
2620 IF(IBMARR.EQ.1,OR,M.EQ.MMAX) GOTO 330
2621 IF(IFIELD(N,M).EQ.KOCN.AND.IFIELD(N,MP).EQ.KOCN) GOTO 330
2622 C CHECK FOR INCLUSION OF NON-LINEAR TERMS:NLT(1=NO;2=YES)
2623 EDGE=AMINI1(FEDGE(N,M),FEDGE(N,MP))
2624 FNOLN=FNOLN0
2625 IF(EDGE.LT.0.55) FNOLN=0.
2626 C CHECK FOR CHANNEL: NO=0, YES=1.
2627 BETAC=1-IFIX(BX(N,M))
2628 C DEPTH-INDEPENDENT VARIABLES
2629 HS=D(N,M)+BETAH*SE(N,M)
2630 HP=D(N,MP)+BETAH*SE(N,MP)
2631 HPP=D(N,MP+1)+BETAH*SE(N,MP+1)
2632 HB=.5*(HS+HP)
2633 HI=1./ (HB+EP)
2634 HISQ=B2*HI**2
2635 C TOTAL EXTERNAL RETARDING FORCES PLUS SIDE STRESS
2636 CFS=DTI*BTAC*CDRGWS/(DL*BX(N,M))
2637 XDT=DTI*(TBX(N,M)-.5*TSX(N,M)-.5*TSX(N,MP))*HI+EDGE*GSTARX(N,M)
2638 1 .4*THETSU(N,M)*UE(N,M)
2639 FCOR=EDGE*FSPLIT*(FCOR0+DFDM*FLOAT(M-MCOR)+DFDN*FLOAT(N-NCOR))
2640 DDEL=.25*(SEPP(N,M)+SEPP(N,MP)-SOLD(N,M)-SOLD(N,MP))*HI
2641 RAP=B2*HI/BX(N,M)* (BX(N,M)+BX(N,MP))
2642 RAM=B3*HI/BX(N,M)* (BX(N,M)+BX(N,MM))
2643 RBP=B4*HI
2644 RBM=B4*HI
2645 FAVG=1./ (FB+FLOAT(NFLUX(N,M)+NFLUX(NM,MP)+NFLUX(N,MP)+NFLUX(NM,MP)))
2646 C NON-LINEAR TERMS
2647 CU=SIGN(1.0,UH(N,M))
2648 IF(FNOLN.EQ.0.) THEN
2649 ANA=0.
2650 ELSE
2651 F1=B7*HI/BX(N,M)
2652 HBXM=F1*BX(N,MM)*(HS+D(N,MM)+BETAH*SE(N,MM))
2653 HBX=F1*BX(N,M)*(HS+HP)
2654 HBXP=F1*BX(N,MP)*(HP+HPP)
2655 HCP=B8*HI*(HS+HP+D(N,MP)+BETAH*(SE(NP,M)+SE(NP,MP)))
2656 HCM=B8*HI*(HS+HP+D(NM,MP)+D(NM,MP)+BETAH*(SE(NM,M)+SE(NM,MP)))
2657 ANA=FNOLN*(B5*H1*BX(N,M)*(1.-CU)*(THETA1(N,MP)*UH(N,MP)))
2658 1 -THETA1(N,M)*UH(N,M))+(1.+CU)*(THETA1(N,M)*UH(N,M))
2659 2 -THETA1(N,MM)*UH(N,MM))+B6*HI*(THETA3(N,M)-THETA3(N,MM)))
2660 END IF
2661 C APPLY TOP BOUNDARY CONDITIONS
2662 IF(ITOPV.EQ.1) THEN
2663 FA(1)=HB*DP/(AV(1,N,M)+AV(1,N,MP))*(TSX(N,M)+TSX(N,MP))
2664 FB(1)=1.-0.
2665 ELSE IF(ITOPV.EQ.2) THEN
2666 FA(1)=B9*HB*DP/(AV(1,N,M)+AV(1,N,MP))*(TSX(N,M)+TSX(N,MP))
2667 FB(1)=B10
2668 ELSE IF(ITOPV.EQ.3) THEN
2669 DP=HISQ*(AV(1,N,M)+AV(1,N,MP))
2670 GB=FNOLN*B1*H1*(W(2,N,M)+W(2,N,MP)+W(1,N,M)+W(1,N,MP))
2671 DIFF=RAP*A3(1,N,MP)*(U(1,N,MP)-U(1,N,M))
2672 1 -RAM*A3(1,N,M)*(U(1,N,M)-U(PMM(1,N)))
2673 2 +(1.-BETAC)*(RBP*(U(1,NP,MP)-U(1,N,M)+V(1,N,MP)-V(1,N,M))
2674 3 * .25*(AH3(1,N,M)+AH3(1,N,MP)+AH3(1,N,MP)+AH3(1,N,MP))
2675 4 -RBM*(U(1,N,M)-U(PMM(1,NM))+V(1,NM,MP)-VPM(1,NM)))
2676 5 * .25*(AH3(1,N,M)+AH3(1,N,MP)+AH3(1,N,MP)+AH3(1,N,MP))
2677 UCEN=U(1,N,M)+UE(N,M)
2678 VB=(VPM(1,N,MP)+V(1,N,MP)+V(1,N,MP)+V(1,N,MP))*FAVG
2679 ANB=FNOLN*((1.-CU)*(HBXP*(U(1,N,MP)+UE(N,MP))*2*-HBX*(UCEN)**2)
2680 1 +(1.+CU)*(HBXP*(U(1,N,MP)+UE(N,MP))*2*-HBX*(UCEN)**2)
2681 2 +HCP*(U(1,NP,MP)+UE(N,M)+UCEN)*(V(1,N,M)+VE(N,M))
2682 3 +(1.,N,M)*V(1,NM,MP)+VE(NM,MP))
2683 4 -V(E(NM,M)*V(1,NM,MP)+VE(NM,MP))
2684 5 * .25*(AH3(1,N,M)+AH3(1,N,MP)+AH3(1,N,MP)+AH3(1,N,MP))
2685 XDT=(1.+DELH-2.*G1+G2+G3+G4+CFS*ABS(UCEN))
2686 DENOM=1./(1.+DELH-2.*G1+G2+G3+G4+CFS*ABS(UCEN))
2687 FA(1)=(UPM(1,N)*V(1,-DELH)+(XDT-CFS*ABS(UCEN))*UE(N,M)
2688 1 -EDGE*DTI*GRX(1)+FCR*VB+DIFF+ANA-ANB)+XDT1)*DENOM
2689 FB(1)=2.* (GDP+DP)*DENOM
2690 ENDIF
2691 C
2692 180 DO 220 L=2,LAYRS
2693 DP=HISQ*(AV(L,N,M)+AV(L,N,MP))
2694 DM=HISQ*(AV(L-1,N,M)+AV(L-1,N,MP))
2695 VB=(VPM(L,N)+VPM(L,MP)+V(L,N,M)+V(L,N,MP))*FAVG
2696 C MOMENTUM DIFFUSION TERMS
2697 DIFF=RAP*A3(L,N,MP)*(U(L,N,MP)-U(L,N,M))
2698 1 -RAM*A3(L,N,M)*(U(L,N,M)-U(PMM(L,N)))
2699 2 +(1.-BETAC)*(RBP*(U(L,NP,MP)-U(L,N,M)+V(L,N,MP)-V(L,N,M))
2700 3 * .25*(AH3(L,N,M)+AH3(L,N,MP)+AH3(L,N,MP)+AH3(L,N,MP))
2701 4 -RBM*(U(L,N,M)-U(PMM(L,NM))+V(L,NM,MP)-VPM(L,NM)))
2702 5 * .25*(AH3(L,N,M)+AH3(L,N,MP)+AH3(L,N,MP)+AH3(L,N,MP))
2703 C NON-LINEAR TERMS
2704 UCEN=U(L,N,M)+UE(N,M)
2705 ANB=FNOLN*((1.-CU)*(HBXP*(U(L,N,MP)+UE(N,MP))*2*-HBX*(UCEN)**2)
2706 1 +(1.+CU)*(HBXP*(U(L,N,MP)+UE(N,MP))*2*-HBX*(UCEN)**2)
2707 2 +HCP*(U(L,NP,MP)+UE(N,M)+UCEN)*(V(L,N,M)+VE(N,M))
2708 3 +(1.,N,M)*V(L,NM,MP)+VE(NM,MP))
2709 4 -HCP*(UCEN*UDPM(L,N,M)+UE(N,M)*V(L,N,MP)+VE(N,M))
2710 5 *(VPM(L,N,M)+VE(N,M)*V(L,N,MP)+VE(NM,MP)))
2711 GM=FNOLN*B1*H1*(W(L-1,N,M)+W(L-1,N,MP)+W(L,N,M)+W(L,N,MP))
2712 GB=FNOLN*B1*H1*(W(L+1,N,M)+W(L+1,N,MP)+W(L,N,M)+W(L,N,MP))
2713 C CALCULATE THE RECURSIVE TERMS
2714 F1=GM-DM
2715 F2=1.+DELH+GM+DP+DM+CFCS*ABS(UCEN)
2716 F3=GP-DP
2717 F4=UPM(L,N)*(1.-DELH)+XDT-CFS*ABS(UCEN)*UE(N,M)-EDGE*DTI*GRX(L)
2718 1 +FCR*VB+DIFF+ANA-ANB
2719 DENOM=1./ (F2+F1*FB(L-1))
2720 FA(L)=(F4-F1*FA(L-1))*DENOM
2721 FB(L)=(FBC(L)*F1-F3)*DENOM
2722 220 CONTINUE
2723 C APPLY BOTTOM BOUNDARY CONDITIONS
2724 IF(IBOTV.EQ.0) THEN
2725 UP(LBOT)=UE(N,M)
2726 ELSE IF(IBOTV.EQ.1) THEN ! FIRST-ORDER
2727 RR=DQ*HB*PH(N,M,1,MP)/(AV(LAYRS,N,M)+AV(LAYRS,N,MP))
2728 UP(LBOT)=RR*(PH(N,M,1,MP)-PH(N,M,0,MP))/RR
2729 ELSE IF(IBOTV.EQ.2) THEN
2730 RR=DQ*HB*(GHI(N,M)+PHI(N,MP))/(AV(LAYRS,N,M)+AV(LAYRS,N,MP))
2731 1 -(L-1)*FA(LAYRS))/((4.-FB(LAYRS-1))*FB(LAYRS)-3.-2.*RR)
2732 2 +RR*UE(N,M)+FA(LAYRS-1)*FB(LAYRS-3,-2.*RR)
2733 3 -RR*UE(N,M)+FA(LAYRS-1)*FB(LAYRS-3,-2.*RR)
2734 4 -.25*(PH(N,M)+PHI(N,MP))
2735 5 -.50*(AV(LAYRS,N,M)+AV(LAYRS,N,MP))/ (DQ*HB)
2736 UP(LBOT)=(FA(LAYRS)*(F2-F1)-2.*F1*UE(N,M))/(F1+F2)
2737 1 -FB(LAYRS)*(F2-F1))
2738 ENDIF
2739 C FIND UPDATED VELOCITY, AND NET VELOCITY
2740 270 UNET=HALFDQ*UP(LBOT)
2741 DO 280 I=1,LAYRM
2742 UP(LBOT-I)=FA(LBOT-I)+FB(LBOT-I)*UP(LBOT-I+1)
2743 280 UNET=UNET+DQ*UP(LBOT-I)
2744 C SECOND ORDER

```

```

2745 290 GOTO(290,300,290),ITOPV
2746 UP(1)=FA(1)+FB(1)*UP(2)
2747 GOTO 310
2748 300 UP(1)=FB(1)*UP(2)-.33333*UP(3)+FA(1)
2749 310 UNET=UNET+HALFD0*UP(1)
2750 C ENFORCE ZERO NET INTERNAL-MODE FLOW
2751 DO 320 L=1,LBOT
2752 320 U(L,N,M)=UP(L)-UNET*CJ(L)
2753 C Y-DIRECTION INTERNAL MODE VELOCITY
2755 C
2756 330 IF(IBARR,GE,2)GOTO 540
2757 IF(N.EQ.NMAX)GOTO 540
2758 IF(IFIELD(N,M).EQ.KOQN.AND.IFIELD(NP,M).EQ.KOCN)GOTO 540
2759 C CHECK FOR NON-LINEAR TERMS
2760 EDGE=AMINI(FEDGE(N,M),FEDGE(NP,M))
2761 FNOLN=FNOLN0
2762 IF(EDGE,LT,0.55)FNOLN=0.
2763 C CHECK FOR CHANNEL: NO=0, YES=1.
2764 BETAC=-1#IX(BY(N,M))
2765 C DEPTH VARIABLES
2766 HS=H(N,M)+BETAH*SE(N,M)
2767 HD=H(N,M)+BETAH*SE(NP,M)
2768 HBP=HP-
2769 IF(NP,LT,1,MAX)HPP=D(NP+1,M)+BETAH*SE(NP+1,M)
2770 HB=.5*(HS+HP)
2771 HI=1./((HD+HP)
2772 CFS=DTI*BETAC*CDRGWS/(DL*BY(N,M))
2773 C TOTAL EXTERNAL RETARDING FORCE
2774 YDT=DTI*((TB(Y,N,M)-.5*(TSY(N,M)+TSY(NP,M)))*HI+EDGE*GSTARY(N,M)
2775 1 + (THETSV(N,M))*VE(N,M))
2776 HISB=B2*H**2
2777 DELH=.25*(SEPP(N,M)+SEPP(NP,M)-SOLD(N,M)-SOLD(NP,M))*HI
2778 FAVG=1./((E#FLOAT(MFLUX(N,M))+FLOAT(MFLUX(N,MM)))
2779 1 +FLOAT(MFLUX(NP,M))+FLOAT(MFLUX(NP,MM)))
2780 FCOR=EDGE*FSPLIT*(FCRCR+DFDM*FLOAT(M-MCOR)+FDNN*FLOAT(N-NCOR))
2781 RAP=B3*HI/BY(N,M)*BY(N,M)+BY(NP,M)
2782 RAM=B3*HI/BY(N,M)*BY(N,M)+BY(NP,MM)
2783 RBP=B4*HI
2784 RBB=B4*HI
2785 C NON-LINEAR TERMS
2786 CV=SIGN(1.0,VH(N,M))
2787 IF(FNOLN.EQ.0.)THEN
2788 ANA=0.
2789 ELSE
2790 F1=B7*HI/BY(N,M)
2791 HBYM=F1*BY(N,M)*(HS+D(NM,M)+BETAH*SE(NM,M))
2792 HBY=F1*BY(N,M)*(HS+HP)
2793 HBYP=F1*BY(NP,M)*(HP+HPP)
2794 HCP=B8*HI*(HS+HP+D(N,MP)+D(NP,MP)+BETAH*(SE(N,M)+SE(NP,MP)))
2795 HCM=B8*HI*(HS+HP+D(N,MM)+D(NP,MM)+BETAH*(SE(N,MM)+SE(NP,MM)))
2796 ANA=B5*HI/BY(N,M)*(1.-CV)*(THETA2(N,M)*VH(N,M)
2797 1 -THETA2(N,M)*VH(N,M)+(1.+CV)*(THETA2(N,M)*VH(N,M)
2798 2 -THETA2(NM,M)*VH(NM,M))+B6*HI*(THETA3(N,M)-THETA3(N,MM))
2799 ENDIF
2800 C APPLY TOP BOUNDARY CONDITIONS
2801 IF(ITOPV,EQ,1)THEN
2802 FA(1)=HB*D/(AV(1,N,M)+AV(1,NP,M))*(TSY(N,M)+TSY(NP,M))
2803 FB(1)=1.0
2804 ELSE IF(ITOPV,EQ,2)THEN
2805 FA(1)=.66667*HB*D/(AV(1,N,M)+AV(1,NP,M))*(TSY(N,M)+TSY(NP,M))
2806 FB(1)=1.3333
2807 ELSE IF(ITOPV,EQ,3)THEN
2808 D=HISB*AV(1,N,M)+AV(1,NP,M)
2809 GB=FNOLN*(B1*(W(2,N,M)+W(Z,NP,M)+W(1,N,M)+W(1,NP,M))
2810 DIFF=DP*(B1*(W(1,N,M)+W(1,NP,M)-V(1,N,M)
2811 1 -RAM)*AH3(1,N,M)+AH3(1,NP,M)-VEM(1,N,M))
2812 2 +(1.-BETAH)*(BSP*(U(1,N,M)-UPM(1,N)+V(1,N,MP)-V(1,N,M))
2813 3 *.25*(AH3(1,N,M)+AH3(1,NP,M)+AH3(1,N,MP)+AH3(1,NP,MP))
2814 4 -RBM*(UPM(1,N)-UPRM(1,N)+V(1,N,M)-VPM(1,N)
2815 5 *.25*(AH3(1,N,M)+AH3(1,NP,M)+AH3(1,N,MM)+AH3(1,NP,MM)))
2816 VCEN=V(1,N,M)+VE(N,M)
2817 UB=(UPM(1,N)+DPM(1,N)+UPMM(1,NP)+UEMM(1,NP))*FAVG
2818 ANB=FNOLN*((1.-CV)*(HBY*V(1,NP,M)+VE(NP,M))*2*-HBY*(VCEN)**2)
2819 1 +(1.+CV)*(HBY*(VCEN)**2-HBYM*(VPM(1,NM)+VE(NM,M))*2)
2820 2 +HCP*(U(1,N,M)+UB(1,N,M)+U(1,NP,M)+UE(NP,M))*VCEN
2821 3 +V(1,N,MP)+VE(NP,M))*RCM*(UPM(1,N)+UE(N,MM)+UPMM(1,NP)
2822 4 +UE(NP,MM))*VCEN+VEM(1,N)+VE(N,MM))
2823 YDT=.5*DT*(TSY(N,M)+TSY(NP,M))*HI*DQ
2824 DENOM=1./((L+DELH+2.*(DP-GP)+CFS*ABS(VCEN))
2825 FA(1)=(VPM(1,N)*(1.-DELI)+(YDT-CFS*ABS(VCEN))*VE(N,M)
2826 1 -EDGE*DTI*GRY(1)-FCOR*UB+DIFF+ANA-ANB)+YDT1)*DENOM
2827 FB(1)=2.*GP+DP)*DENOM
2828 ENDIF
2829 C
2830 DO 430 L=2,LAYRS
2831 390 DO 430 L=2,LAYRS
2832 DB=HISB*(AV(L,N,M)+AV(L,NP,M))
2833 DM=HISB*(V(L,N,M)+V(L,NP,M)+AN(L-1,N,M))
2834 UB=(UPM(L,N)+UM(L,N)+UPM(L,NP)+UPMM(L,NP))*FAVG
2835 C HORIZONTAL DIFFUSION OF MOMENTUM
2836 DIFF=RAP*AH3(L,NP,M)*V(L,NP,M)-V(L,N,M)
2837 1 -RAM*AH3(L,N,M)*V(L,N,M)-VEM(L,NM)
2838 2 +(1.-BETAH)*(RBP*(U(L,NP,M)-UPM(L,N)+V(L,N,MP)-V(L,N,M))
2839 3 *.25*(AH3(L,N,M)+AH3(L,NP,M)+AH3(L,N,MP)+AH3(L,NP,MP))
2840 4 -RBM*(UPM(L,NP)-UPPM(L,N)+V(L,N,M)-VPM(L,N))
2841 5 *.25*(AH3(L,N,M)+AH3(L,NP,M)+AH3(L,N,MM)+AH3(L,NP,MM)))
2842 C NON-LINEAR TERMS
2843 VCEN=V(1,N,M)+VE(N,M)
2844 ANB=FNOLN*((1.-CV)*(HBY*V(1,NP,M)+VE(NP,M))*2*-HBY*(VCEN)**2)
2845 1 +(1.+CV)*(HBY*(VCEN)**2-HBYM*(VPM(1,NM)+VE(NM,M))*2)
2846 2 +HCP*(U(1,N,M)+UB(1,N,M)+U(1,NP,M)+UE(NP,M))*VCEN
2847 3 *(VCEN+V(L,N,M)+VE(N,MM))
2848 4 -HCM*(UPM(1,N)+UE(N,MM)+UPM(1,NP)+UE(NP,MM))
2849 5 *(VCEN+VEM(1,N)+VE(N,MM))
2850 GM=FNOLN*BI*HI*(W(L-1,N,M)+W(L-1,MP)+W(L,N,M)+W(L,NP,M))
2851 GP=FNOLN*BI*HI*(W(L-1,N,M)+W(L+1,NP,M)+W(L,N,M)+W(L,NP,M))
2852 C COMPUTE THE RECURSIVE ARRAYS
2853 F1=GM-DM
2854 FS=CFS*ABS(VCEN)
2855 F2=1.+DH*GM-GP+DM+DP+FS
2856 F3=DP-DI+FS
2857 F4=VPM(L,N)*(1.-DELI)+YDT-FS*VE(N,M)-EDGE*DTI*GRY(L)-FCOR*UB+DIFF
2858 1 +ANA-ANB
2859 DENOM=1./((P2+F1*FB(L-1))
2860 FA(L)=(F4-F1*FA(L-1))*DENOM
2861 FB(L)=(FBC(L)*F1-F3)*DENOM
2862 430 CONTINUE
2863 C APPLY BOTTOM BOUNDARY CONDITIONS
2864 IF(IBOTV,EQ,0)THEN
2865 VP(LBOT)=VE(N,M)
2866 ELSE IF(IBOTV,EQ,1)THEN ! FIRST ORDER
2867 RR=DQ*HB*(PHI(N,M)+PHI(NP,M))/(AV(LAYRS,N,M)+AV(LAYRS,NP,M))
2868 VP(LBOT)=(FA(LAYRS)-RR*VE(N,M))/(1.+RR-FB(LAYRS))
2869 ELSE IF(IBOTV,EQ,2)THEN ! SECOND ORDER
2870 RR=DQ*HB*(PHI(N,M)+PHI(NP,M))/(AV(LAYRS,N,M)+AV(LAYRS,NP,M))
2871 VP(LBOT)=(2.*RR*VE(N,M)+FA(LAYRS-1)-(4.-FB(LAYRS
2872 1 -1.))*FA(LAYRS))/(4.-FB(LAYRS-1))*FB(LAYRS-3.-2.*RR)
2873 ELSE IF(IBOTV,EQ,3)THEN ! STRESS AT MID-LEVEL
2874 F1=.25*(PHI(N,M)+PHI(NP,M))
2875 F2=.50*(AV(LAYRS,N,M)+AV(LAYRS,NP,M))/(DQ*HB)
2876 VP(LBOT)=(FA(LAYRS)*(F2-F1)-2.*F1*VE(N,M))/(F1+F2)
2877 1 -FB(LAYRS)*(F2-F1))
2878 END IF
2879 C FIND UPDATED VELOCITY AND NET VELOCITY
2880 480 UNET=HALFD*VP(LBOT)
2881 DO 490 I=1,LAYRM
2882 VP(LBOT-I)=FA(LBOT-I)+FB(LBOT-I)*VP(LBOT-I+1)
2883 490 UNET=UNET+DQ*VP(LBOT-I)
2884 GOTO(500,150,500),ITOPV
2885 500 VP(1)=FA(1)+FB(1)*VP(2)
2886 GOTO 520
2887 510 VP(1)=FB(1)*VP(2)-.33333*VP(3)+FA(1)
2888 520 UNET=HALFD*VP(1)
2889 C ENFORCE ZERO NET FLO
2890 DO 530 L=1,LBOT
2891 530 V(L,N,M)=VP(L)-UNET*CJ(L)
2892 C
2893 540 CONTINUE
2894 545 CONTINUE
2895 550 CONTINUE
2896 C
2897 560 CONTINUE
2898 RETURN
2899 END
2900 C
2901 C -----
2902 C
2903 SUBROUTINE VERVIS
2904 C SEPTEMBER 1987 MEAD K.W.HESS VAX 11/780
2905 C PURPOSE - TO UPDATE ALL VERTICAL EXCHANGE COEFFICIENTS.
2906 C USE THE MONK AND ANDERSON FORMULATION.
2907 C VARIABLES -
2908 C INDEX = RUN UPDATE INDEX:0=RESTART CONDITION,1=NORMAL
2909 C RICHNO = RICHARDSON NUMBER = (D(RHO)/DZ)/(D(DZ)**2)*RHO*G
2910 INCLUDE 'COMMON.FOR'
2911 COMMON/AI/DTIMAX
2912 BETAH=MAX0(0,MIN0(1,IBETAH))
2913 IF(INSTL.EQ.1)DTIMAX=1.E+10
2914 C SET VERTICAL DENSITY CHANGE (GM/CC) PER METER
2915 DO 550 M=1,MAXM
2916 DO 550 N=1,NMAX
2917 IF(IFIELD(N,M).LE.0)GOTO 120
2918 AVMAX=0.0
2919 MM=MAX0(M-1,1)
2920 MP=MIN0(M+1,MMAX)
2921 NM=MAX0(N-1,1)
2922 NP=MIN0(N+1,MMAX)
2923 HS=AMAX1(1,0,D(N,M)+BETAH*SE(N,M))
2924 F1=AG/(HS*DQ)
2925 F2=1./(HS*DQ)
2926 C ESTABLISH WEIGHTING FACTORS FOR VELOCITY
2927 FD=1./AMAX1(1,414,FLOAT(MFLUX(N,M)+MFLUX(N,MM)))
2928 FX1=FLOAT(MFLUX(N,M))+FD
2929 IF(M.EQ.1)FX1=1.0
2930 FX2=FLOAT(MFLUX(N,MM))+FD
2931 FD=1./AMAX1(1,414,FLOAT(NFLUX(N,M)+NFLUX(N,MM)))
2932 FY1=FLOAT(NFLUX(N,M))+FD
2933 IF(N.EQ.1)FY1=1.0
2934 FY2=FLOAT(NFLUX(NM,M))+FD
2935 C LOOP THRU LAYERS
2936 DO 550 L=1,LAYRS
2937 FAV=0.
2938 FDV=0.
2939 Q=ABS(FLOAT(1-L)*DQ-HALFDQ)
2940 C mixing length for momentum diffusivity
2941 Z1=VONKAR*HS*(1.-Q)*CRO
2942 C GET VELOCITY GRADIENTS
2943 DUM=U(L,N,MM)-U(L+1,N,MM)
2944 IF(M.EQ.1)DUM=0.0
2945 DVM=V(L,N,MM)-V(L+1,MM,MM)
2946 IF(N.EQ.1)DVM=0.0
2947 DELI=FX1*(U(L,N,M)-U(L+1,N,M))+EX2*(DVM)
2948 DELV=FY1*(V(L,N,M)-V(L+1,N,M))+FY2*(DVM)
2949 DUDZ=SQRT(DELU**2+DELV**2)*F2+E
2950 C GET DENSITY GRADIENT PER UNIT DENSITY
2951 IF(ICOUPL,GT,0)THEN
2952 DELTA=FRHO(S(L,N,M),T(L,N,M))-FRHO(S(L+1,N,M),T(L+1,N,M))
2953 ELSE
2954 DELRH0=0.0000
2955 DELTA=DELRH0*HS*DQ
2956 END IF
2957 C NEW RICHARDSON NUMBER: ALLOWS FOR NEGATIVE RI
2958 RICHNO=RAMP*F1/DELTA/(DUDZ**2)
2959 RICHNO=MIN0(RIMIN,AMIN1(RIMAX,RICHNO))
2960 R1(L,N,M)=RICHNO
2961 IF(RICHNO,GT,0.0)THEN
2962 FAV=CRICH(1)*(1.+CRICH(2)*RICHNO)**(-CRICH(3))
2963 FDV=CRICH(5)*(1.+CRICH(6)*RICHNO)**(-CRICH(7))
2964 ELSE
2965 FAV=CRICH(1)*(1.+CRICH(4)*RICHNO**2)
2966 FDV=CRICH(5)*(1.+CRICH(8)*RICHNO**2)
2967 ENDIF
2968 90 VISNEW=AV0*FAV*DUDZ*21**2
2969 DIFNEW=DVO+FDV*DUDZ*21**2
2970 C UPDATE
2971 IF(AV(L,N,M).EQ.0.0)THEN
2972 AV(L,N,M)=VISNEW
2973 ELSE
2974 AV(L,N,M)=SQRT(AV(L,N,M)*VISNEW)
2975 END IF
2976 IF(DV(L,N,M).EQ.0.0)THEN
2977 DV(L,N,M)=DIFNEW
2978 ELSE
2979 DV(L,N,M)=SQRT(DV(L,N,M)*DIFNEW)
2980 END IF
2981 100 IF(L.GT.1)AVMAX=AMAX1(AVMAX,AV(L,N,M))
2982 C UPDATE SCALE EXPLICIT INTERNAL-MODE TIMESTEP BASED ON DIFFUSION
2983 IF(HR,GT,HRCON2.AND.AVMAX.GT.0.0)DTIMAX=AMIN1(DTIMAX,
2984 1 -.25*(DQ*HS)**2/AVMAX)
2985 120 CONTINUE
2986 130 CONTINUE
2987 C LOOP THRU OCEANIC BOUNDARIES
2988 IF(NUMBC,L,0)RETURN
2989 DO 200 IB=1,NUMBC
2990 DO 200 N=NB1(IB),NB2(IB)
2991 DO 200 M=M#1(IB),MB2(IB)
2992 MP=M
2993 IF(IABS(ITPO(IB)).EQ.1)MP=M-ISIGN(1,ITPO(IB))
2994 NP=N
2995 IF(IABS(ITPO(IB)).EQ.2)NP=N-ISIGN(1,ITPO(IB))
2996 DO 1,LAYRS

```

```

2997      AV(L,N,M)=AV(L,NP,MP)
2998      DV(L,N,M)=DV(L,NP,MP)
2999      ENDDO
3000 200 CONTINUE
3001  IF(IPRNT1.GT.0)WRITE(ISCR,*)'UT=',UT,' DTIMAX=',DTIMAX
3002  IF(IPRNT1.GT.0)WRITE(ISCR,*)'AVMAX=',AVMAX
3003  RETURN
3004  END
3005 C-----+
3006 C-----+
3007 C-----+
3008 SUBROUTINE WVERT
3009 C      MARCH 1986 K.W. HESS MEAD VAX 11/750
3010 C      PURPOSE - TO CALCULATE W, THE PRODUCT OF H AND DQ/DTI
3011 C      INCLUDE 'COMM20.FOR'
3012 C      SET CONSTANTS
3013 F1=.25*DQ/DL
3014 KOCN=10*KOCNbc
3015 C      LOOP THRU THE GRID MESH
3016 DO 140 M=1,MMAX
3017 NA=M+1(M)/1000
3018 NB=NAB(M)-1000*NA
3019 MM=MAX(1,M)
3020 MB=MIN(1,M,MMAX)
3021 IF(NA.GT.NB)GOTO 140
3022 DO 130 N=NR,NB
3023 IF(IFIELD(N,M)/10.LT.1.OR.IFIELD(N,M)/10.EQ.KOCNbc)GOTO 130
3024 C      WATER GRID HERE
3025 HC=D(N,M)+SE(N,M)
3026 F2=F1/AREA(N,M)
3027 HNM=F2*B(X(N,M))*(HC+D(N,MP)+SE(N,MP))
3028 HNM=F2*B(X(N,MM))*(HC+D(N,MM)+SE(N,MM))
3029 NP=MIN0(N+1,MMAX)
3030 NM=MAX0(N-1,1)
3031 HNP=F2*B(Y(N,M)*(HC+D(NP,M)+SE(NP,M))
3032 HNM=F2*B(Y(NM,M)*(HC+D(NM,M)+SE(NM,M))
3033 C      WC IS THE PRODUCT OF THE DEPTH AND THE DIMENSIONLESS VERT. VEL.
3034 WC(LBOT)=0.0
3035 DO 100 L=LAYRS,1,-1
3036 100 WC(L)=WC(L-1)-(HMP*(U(L,N,M)+U(L+1,N,M))-HMM*(U(L,N,MM)
3037 1 +U(L+1,N,MM))+HNK*(V(L,N,M)+V(L+1,N,M))-HNM*(V(L,NM,M)
3038 2 +V(L+1,NM,M)))
3039 C      GET THE ADJUSTED DIMENSIONLESS VERTICAL VELOCITY
3040 DO 110 L=1,LBOT
3041 110 W(L,N,M)=WC(L)-WC(1)*FLOAT(LBOT-L)*DQ
3042 130 CONTINUE
3043 140 CONTINUE
3044 RETURN
3045 END
3046 C-----+
3047 C      MECCA: FORCES
3048 C-----+
3049 C-----+
3050 SUBROUTINE FORCES
3051
3052 C      APRIL 1986 K. W. HESS MEAD VAX11/750
3053 C      PURPOSE - TO SET THE INTERFACIAL STRESS TERMS:
3054 C      BOTTOM STRESS AND WIND STRESS AND AIR TEMPERATURE
3055 C      INCLUDE 'COMM20.FOR'
3056 C      GET WIND STRESS AND AIR TEMPERATURE
3057 IF(NSTE.EQ.1)CALL ATOMS
3058 C      UPDATE BOTTOM STRESS
3059 CALL BSTRES
3060 C      UPDATE EXTERNAL PRESSURE GRADIENTS
3061 CALL ALGRAD
3062 RETURN
3063 END
3064 C-----+
3065 C-----+
3066 C-----+
3067 SUBROUTINE ATMOS
3068 C      APRIL 1996 K. W. HESS CEOB SGI/IRIS
3069 C      PURPOSE - TO SET THE WIND STRESS AND AIR TEMPERATURE
3070 C      VARIABLES -
3071 INCLUDE 'COMM20.FOR'
3072 COMMON/METOX/TMET8(2),ITYPE1
3073 REAL*8 TMET8,FA8,FB8
3074 RAMPW=AMINI(1.0,CUMDAY)
3075 C      initialize
3076 IF(NSIGW.GT.0.AND.IENDWN.EQ.0)GOTO 100
3077 DPADX=0.0
3078 DPADY=0.0
3079 DO N=1,NMAX
3080 DO M=1,MMAX
3081 TSX(N,M)=0
3082 TSY(N,M)=0
3083 ENDDO
3084 ENDDO
3085 IF(NSIGW.EQ.0)RETURN
3086 IF(IENDWN.EQ.1)THEN
3087 WRITE(6,*)' NO MORE GRIDDED WIND DATA'
3088 RETURN
3089 ENDIF
3090 C      check for time of available data
3091 100 CONTINUE
3092 IF(IPRNT1.EQ.1)WRITE(6,105)TMET8(1),YT,TMET8(2)
3093 105 FORMAT(5X,'TMET8(1)',F12.7,' YT',F12.7,' TMET8(2)',F12.7)
3094 110 IF(YT.GT.TMET8(2))THEN
3095 C      save
3096 TMET8(1)=TMET8(2)
3097 DO N=1,NMAX
3098 DO M=1,MMAX
3099 FX(1,N,M)=FX(2,N,M)
3100 FY(1,N,M)=FY(2,N,M)
3101 ENDDO
3102 ENDDO
3103 C      set default values
3104 DO N=1,NMAX
3105 DO M=1,MMAX
3106 FX(2,N,M)=0
3107 FY(2,N,M)=0
3108 ENDDO
3109 ENDDO
3110 C      read next array
3111 I=2
3112 CALL RDWIN(I,IEND)
3113 IF(IEND.EQ.1)RETURN
3114 IF(IPRNT1.EQ.1)WRITE(6,105)TMET8(1),YT,TMET8(2)
3115 GOTO 110
3116 ENDIF
3117 C      get interpolated stress
3118 FB8=(YT-TMET8(1))/(TMET8(2)-TMET8(1))
3119 F2=FB8
3120 FA8=1.-FB8
3121 F1=FA8
3122 IF(IPRNT1.EQ.1)WRITE(6,*)
3123      F1,2, ITYPE1=' ,F1,F2,ITYPE1
3124      DO M=1,MMAX
3125      TSX(N,M)=RAMPW*(F1*FX(1,N,M)+F2*FX(2,N,M))
3126      TSY(N,M)=RAMPW*(F1*FY(1,N,M)+F2*FY(2,N,M))
3127 IF(ITYPE1.EQ.1)THEN
3128 W1=TSX(N,M)
3129 W2=TSY(N,M)
3130 W10=SQRT(W1**2+W2**2)
3131 CDRGAW=CDR1+CDR2*W10
3132 TSX(N,M)=DENRAT*CDRGAW*W10*W1
3133 TSY(N,M)=DENRAT*CDRGAW*W10*W2
3134 WX(N,M)=W1
3135 WY(N,M)=W2
3136 ENDIF
3137 ENDDO
3138 ENDDO
3139 C      adjust atmospheric pressure gradient
3140 DPADX=RAMPW*DPADX
3141 DPADY=RAMPW*DPADY
3142 C      adjust for shallow water
3143 IF(DPADX.LT.0.0)GOTO 130
3144 DO 120 M=1,MMAX
3145 DO 120 N=1,NMAX
3146 IF(MFLUX(N,M).EQ.1)THEN
3147 DEFF=AMINI(D(N,M)+SE(N,M),D(N,M+1)+SE(N,M+1))
3148 FACTOR=AMAX1(0.0, AMINI(1., (DEFF-DTAU1)/(DTAU2-DTAU1)))
3149 TSX(N,M)=TSY(N,M)*FACTOR
3150 ENDIF
3151 IF(NFLUX(N,M).EQ.1)THEN
3152 DEFF=AMINI(D(N,M)+SE(N,M),D(N+1,M)+SE(N+1,M))
3153 FACTOR=AMAX1(0.0, AMINI(1., (DEFF-DTAU1)/(DTAU2-DTAU1)))
3154 TSY(N,M)=TSY(N,M)*FACTOR
3155 ENDIF
3156 120 CONTINUE
3157 130 CONTINUE
3158 RETURN
3159 END
3160 C-----+
3161 C-----+
3162 C-----+
3163 SUBROUTINE BSTRES
3164 C      SEPTEMBER 1996 K. W. HESS
3165 C      PURPOSE - TO UPDATE THE BOTTOM STRESS. TBX IS AT LOCATION
3166 C      OF UH, TBY IS AT VH.
3167 C-----+
3168 C      VARIABLES
3169 C      IBOTV = BOTTOM CONDITION INDEX
3170 C      0 = NON-SLIP
3171 C      1 = SLIP, FIRST ORDER
3172 C      2 = SLIP, SECND ORDER
3173 C      3 = LOG LAYER
3174 INCLUDE 'COMM20.FOR'
3175 DIMENSION UTO(NUO,NSIZE,MSIZE),VTO(NSIZE,MSIZE)
3176 C      SELECT WEIGHTING FACTOR FOR OLD VS. NEW STRESS
3177 FNEW=1.
3178 FOLD=1.-FNEW
3179 C      COMPUTE THE WEIGHTING FACTORS FOR NO-SLIP AND LOG-LAYER STRESS
3180 GAMMA=1.
3181 IF(IBOTV.EQ.0)GAMMA=.0
3182 IF(IBOTV.EQ.3)GAMMA=.5
3183 C      COMPUTE THE TOTAL BOTTOM VELOCITY
3184 DO M=1,MMAX
3185 DO N=1,NMAX
3186 UTO(N,M)=OE(N,M)+GAMMA*U(LBOT,N,M)+(1.-GAMMA)*U(LBOT-1,N,M)
3187 VTO(N,M)=VE(N,M)+GAMMA*V(LBOT,N,M)+(1.-GAMMA)*V(LBOT-1,N,M)
3188 ENDO
3189 ENDO
3190 C      COMPUTE THE EFFECTIVE BOTTOM DRAG COEFFICIENT, PHI
3191 DO 110 M=1,MMAX
3192 DO 110 N=1,NMAX
3193 IF(IFIELD(N,M).LT.10)GOTO 110
3194 MM=MAX0(M-1,1)
3195 NM=MAX0(N-1,1)
3196 C      DRAG COEFFICIENT AT GRID CENTER BASED ON LOCAL VELOCITY
3197 UB=(UTO(N,M)+UTO(N,MM))/(FLOAT(MFLUX(N,M)+MFLUX(N,MM))+E)
3198 VB=(VTO(N,M)+VTO(N,MM))/(FLOAT(NFLUX(N,M)+NFLUX(N,MM))+E)
3199 C      zero bottom velocity
3200 IF(IBOTV.EQ.0)THEN
3201 PHI(N,M)=CDWB1+CDWB2*SQRT(UB**2+VB**2)
3202 C      log layer
3203 ELSE IF(IBOTV.EQ.3)THEN
3204 PHI(N,M)=(0.4 ALOG(.5*D(N,M)*DQ/Z0))**2*SQRT(UB**2+VB**2)
3205 ELSE
3206 PHI(N,M)=CDWB1+CDWB2*SQRT(UB**2+VB**2)
3207 ENDIF
3208 110 CONTINUE
3209 C      STRESS
3210 DO 140 M=1,MMAX
3211 MP=MFLUX(M-1,MMAX)
3212 DO 140 N=1,NMAX
3213 IF(IFIELD(N,M).LT.10)GOTO 140
3214 NP=MIN0(N+1,MMAX)
3215 TNEWX=.5*(PHI(N,M)+PHI(N,MP))*UTO(N,M)
3216 TNEWY=.5*(PHI(N,M)+PHI(NP,M))*VTO(N,M)
3217 C      TIME AVERAGE
3218 TBX(N,M)=POLD*TBX(N,M)+FNEW*TNEWX
3219 TBY(N,M)=POLD*TBY(N,M)+FNEW*TNEWY
3220 140 CONTINUE
3221 150 CONTINUE
3222 RETURN
3223 END
3224 C-----+
3225 C-----+
3226 C-----+
3227 SUBROUTINE ALGRAD
3228 C      APRIL 1988 HESS MEAD VAX
3229 C      PURPOSE - TO SET ALL HORIZONTAL DENSITY GRADIENTS
3230 INCLUDE 'COMM20.FOR'
3231 C      SET THE INTERNAL PRESSURE RAMP FACTOR
3232 RAMP=AMAX1(.0,AMINI(1.,(HR1-HRC01)/(HRC02-HRC01)))
3233 C      SET INITIAL DENSITY GRADIENTS
3234 C      IP(KONGEN,EQ.0)GOTO 330
3235 DO 320 M=1,MMAX
3236 DO 320 N=1,NMAX
3237 IF(IFIELD(N,M).LT.10)GOTO 320
3238 IBARR=MOD(IFIELD(N,M),10)
3239 IF(IBARR.EQ.3)GOTO 320
3240 CALL GRADP(N,M,1)
3241 320 CONTINUE
3242 330 CONTINUE
3243 RETURN
3244 END
3245 C=====+
3246 C      MECCA FILE : MREAD
3247 C-----+
3248 C-----+

```

```

3249      SUBROUTINE READZ
3250
3251 C      APRIL 1988 K. W. HESS MEAD VAX 11/750
3252 C      PURPOSE - TO READ IN RUN-TIME FILE NAMES IN AN INTERACTIVE
3253 C      MODE
3254 C      INCLUDE 'COMM20.FOR'
3255 C      ZERO OUT INITIAL ARRAYS
3256 C      OPEN FILE AND READ THE CONTROL FILE
3257 C      LUCON=LUKB
3258 C      RAD=PI/180.
3259 C      DENRAT=RHOA/RHOW
3260 C      CALL RDCON(JTEST,IVER)
3261 C      CALL RDCONZ(JTEST,IVER)
3262 C      CALL RDCON(JTEST,IVER)
3263 C      OPEN OUTPUT PRINT FILE
3264 C      CALL FOPEN(IO,FPRINT)
3265 C      OPEN GRAPHING FILES
3266 C      IF(IGPH.GT.0)CALL FOPEN(LUGRF,FGRAPH)
3267 C      OPEN GEOGRAPHY FILE
3268 C      CALL FOPEN(LUCON,FGEO)
3269 C      CALL RDGEOLUCON)
3270 C      CLOSE(LUCON)
3271 C      GET INITIAL CONDITIONS
3272 C      NSTI=0
3273 C      NSTE=0
3274 C      NSTIT=0
3275 C      NSTET=0
3276 C      HRO=0.0
3277 C      UTO=UTO
3278 C      YEAR1=YEAR
3279 C      IF(ICS.EQ.1)THEN
3280 C      CALL RDICS
3281 C      HRO=FLOAT(NSTET)*DTE/3600.
3282 C      CUMDAY=HRO/24.
3283 C      NSTIT=NSTET/ISPLIT
3284 C      ENDIF
3285 C      RETURN
3286 C
3287 C
3288 C
3289 C-----+
3290 C
3291      SUBROUTINE RDCON1(JTEST,IVER)
3292 C      OCTOBER 1984 K. HESS MEAD VAX 11/70 (REV 8-89)
3293 C      PURPOSE - TO READ IN THE CONTROL FILE
3294 C      INCLUDE 'COMM20.FOR'
3295 C      DIMENSION MXNX(NPMSIZ)
3296 C      THIS IS FOR CON FILE VERSIONS 4 - 5 (8-89)
3297 C      JVER1=4
3298 C      JVER2=5
3299 C      IUNIT=LUCON
3300 C      READ DATA FILE HERE
3301 C      READ(IUNIT,100)CTITLE
3302 C      100 FORMAT(8A10)
3303 C      WRITE(ISCR,101)CTITLE
3304 C      101 FORMAT(5X,'CON FILE TITLE: ',8A10)
3305 C      READ(IUNIT,103)FGE0
3306 C      103 FORMAT(A40)
3307 C      READ(IUNIT,*)IVER,JTEST,KTEST
3308 C      WRITE(ISCR,*)IVER,JTEST,KTEST
3309 C      WRITE(ISCR,105)FGE0
3310 C      105 FORMAT(5X,'GEOGRAPHY FILE: ',A40)
3311 C
3312 C      READ MODEL CONFIGURATION DATA
3313 C      IF(JTEST.EQ.1)WRITE(ISCR,108)
3314 C      108 FORMAT(1X,'MODEL CONFIGURATION PARAMETERS')
3315 C      READ(IUNIT,1002)
3316 C      1001 FORMAT(1X)
3317 C      1002 FORMAT(//,1X)
3318 C      1100 READ(IUNIT,* ,ERR=1108)HRMAX,HROUT,HROUTO,HRSAVE
3319 C      IF(JTEST.EQ.1)PRINT*,HRMAX,HROUT,HROUTO,HRSAVE
3320 C      GOTO 1105
3321 C      1108 WRITE(ISCR,1107)
3322 C      1107 FORMAT(5X,'*** (RDCON) ERROR READING HRMAX, ETC. ***')
3323 C      1109 CONTINUE
3324 C      READ(IUNIT,1001)
3325 C      READ(IUNIT,* ,ERR=1118)DTE,ISPLIT,LAYRS
3326 C      DTI=ISPLIT*DTE
3327 C      IF(JTEST.EQ.1)PRINT*,DTE,ISPLIT,LAYRS
3328 C      GOTO 1119
3329 C      1118 WRITE(ISCR,1117)
3330 C      1117 FORMAT(5X,'*** (RDCON) ERROR READING DTE, ISPLIT, ETC. ***')
3331 C      1119 CONTINUE
3332 C      NDZM=LSIZE-1
3333 C      IF(LAYRS.GE.3.AND.LAYRS.LE.NDZM)GOTO 1122
3334 C      WRITE(ISCR,1120)LAYRS,NDZM
3335 C      1120 FORMAT(1X,'*** ERROR: LAYRS=',I3,' IS NOT BETWEEN 3 AND ',I3)
3336 C      STOP
3337 C      1122 CONTINUE
3338 C      TURBULENCE VARIABLES
3339 C      DHAI=1.0
3340 C      READ(IUNIT,1001)
3341 C      READ(IUNIT,* ,ERR=1217)AH00,AHO,CAH,DHAI,RIMIN,RIMAX
3342 C      IF(JTEST.EQ.1)PRINT*,AH00,AHO,CAH,DHAI,RIMIN,RIMAX
3343 C      GOTO 1220
3344 C      1217 WRITE(ISCR,1218)
3345 C      1218 FORMAT(5X,'*** (RDCON) ERROR READING AHO, CAH ***')
3346 C      1220 READ(IUNIT,* ,ERR=1222)AV00,AV0,(CRICH(I),I=1,4)
3347 C      IF(JTEST.EQ.1)PRINT*,AV00,AV0,(CRICH(I),I=1,4)
3348 C      GOTO 1224
3349 C      1222 WRITE(ISCR,1223)
3350 C      1223 FORMAT(5X,'*** (RDCON) ERROR READING AV0, CRICH1, ETC ***')
3351 C      1224 READ(IUNIT,* ,ERR=1226)DV00,DV0,(CRICH(I),I=5,8)
3352 C      IF(JTEST.EQ.1)PRINT*,DV00,DV0,(CRICH(I),I=5,8)
3353 C      GOTO 1230
3354 C      1226 WRITE(ISCR,1227)
3355 C      1227 FORMAT(5X,'*** (RDCON) ERROR READING DV0 CRICH3, ETC ***')
3356 C      C      UPDATE INTERVALS
3357 C      1230 READ(IUNIT,* ,ERR=1255)IHVISC,IVISC,CRO
3358 C      IF(JTEST.EQ.1)PRINT*,IHVISC,IVISC,CRO
3359 C      GOTO 1234
3360 C      1255 WRITE(ISCR,1256)
3361 C      1256 FORMAT(5X,'*** (RDCON) ERROR READING IHVISC, IVISC. ***')
3362 C      C      DRAG COEFFICIENTS
3363 C      128 READ(IUNIT,1001)
3364 C      READ(IUNIT,* ,ERR=1231)CDWB1,CDWB2,CDRGWS,CDR1,CDR2
3365 C      IF(JTEST.EQ.1)PRINT*,CDWB1,CDWB2,CDRGWS,CDR1,CDR2
3366 C      GOTO 1234
3367 C      1231 WRITE(ISCR,1232)
3368 C      1232 FORMAT(5X,'*** (RDCON) ERROR READING CDWB1, CDWB2,CDRGWS ***')
3369 C      C      HEATING CONSTANTS
3370 C      1234 READ(IUNIT,1001)
3371 C      READ(IUNIT,* ,ERR=1240)ALB,D10PCT
3372 C      IF(JTEST.EQ.1)PRINT*,ALB,D10PCT
3373 C      GOTO 1245
3374 C      1240 WRITE(ISCR,1244)

3375 C      1244 FORMAT(5X,'*** (RDCON) ERROR READING CLOUD, RH ***')
3376 C      SWITCHES
3377 C      1245 READ(IUNIT,1001)
3378 C      READ(IUNIT,* ,ERR=1260)ICOR,IBETAA,IBETAP,IBETAH
3379 C      IF(JTEST.EQ.1)PRINT*,ICOR,IBETAA,IBETAP,IBETAH
3380 C      GOTO 1270
3381 C      1260 WRITE(ISCR,1274)
3382 C      1262 FORMAT(5X,'*** (RDCON) ERROR READING ICOR, ETC. ***')
3383 C      1270 READ(IUNIT,* ,ERR=1272)IEXTRN,INTER,KONCEN,ICOUL
3384 C      IF(JTEST.EQ.1)PRINT*,IEXTRN,INTER,KONCEN,ICOUL
3385 C      GOTO 1280
3386 C      1272 WRITE(ISCR,1274)
3387 C      1274 FORMAT(5X,'*** (RDCON) ERROR READING IEXTRN, ETC. ***')
3388 C      1280 READ(IUNIT,* ,ERR=1285)IOPV,IBOTV,IHEAT,ICPOS
3389 C      IF(JTEST.EQ.1)PRINT*,IOPV,IBOTV,IHEAT,ICPOS
3390 C      GOTO 1290
3391 C      1285 WRITE(ISCR,1287)
3392 C      1287 FORMAT(5X,'*** (RDCON) ERROR READING IHEAT, ETC. ***')
3393 C      1290 CONTINUE
3394 C      READ PRINT PARAMETERS
3395 C      136 CONTINUE
3396 C      IF(JTEST.EQ.1)WRITE(ISCR,135)
3397 C      135 FORMAT(1X,'MODEL PRINT PARAMETERS')
3398 C      READ(IUNIT,1002)
3399 C      PLAN VIEW VARIABLES
3400 C      READ(IUNIT,* ,ERR=137)(JPRNT(I),I=1,13)
3401 C      IF(JTEST.EQ.1)PRINT*,(JPRNT(I),I=1,13)
3402 C      GOTO 1390
3403 C      137 WRITE(ISCR,138)
3404 C      138 FORMAT(5X,'*** (RDCON) ERROR READING JPRNT, ETC. ***')
3405 C      PAGE FORMATS
3406 C      1390 READ(IUNIT,1001)
3407 C      READ(IUNIT,* ,ERR=1392)KPRNT1,KPRNT2
3408 C      IF(JTEST.EQ.1)PRINT*,KPRNT1,KPRNT2
3409 C      GOTO 140
3410 C      1392 WRITE(ISCR,1394)
3411 C      1394 FORMAT(5X,'*** (RDCON) ERROR READING IEXTRN, ETC. ***')
3412 C      PRINT AT ALL LEVELS
3413 C      140 READ(IUNIT,1001)
3414 C      READ(IUNIT,* ,ERR=141)NPRMN
3415 C      IF(JTEST.EQ.1)PRINT*,NPRMN
3416 C      GOTO 144
3417 C      WRITE(ISCR,142)
3418 C      142 FORMAT(5X,'*** (RDCON) ERROR READING NPRMN, ETC. ***')
3419 C      144 READ(NPRN,LE,NPRN)GOTO 146
3420 C      WRITE(ISCR,145)NPRMN,NPRN
3421 C      145 FORMAT(1X,'*** ERROR: NPRMN=',I2,' GREATER THAN NDPRN=',I2)
3422 C      STOP
3423 C      CONTINUE
3424 C      IF(NPRMN.GT.0)READ(IUNIT,*)(IPRMN(J),J=1,NPRMN)
3425 C      IF(NPRMN.GT.0)PRINT*,(IPRMN(J),J=1,NPRMN)
3426 C      CELLS IN LONGITUDINAL SECTION
3427 C      READ(IUNIT,1001)
3428 C      READ(IUNIT,* ,ERR=1400)ISLICE
3429 C      IF(JTEST.EQ.1)PRINT*,ISLICE
3430 C      GOTO 1400
3431 C      1400 WRITE(ISCR,1402)
3432 C      1402 FORMAT(1X,'*** ERROR READING ISLICE ***')
3433 C      1410 IF(ISLICE.LE.NDSL1)GOTO 1406
3434 C      WRITE(ISCR,1404)ISLICE,NDSL1
3435 C      1404 FORMAT(1X,'*** ERROR: ISLICE=',I2,' GREATER THAN NDSL1=',I2)
3436 C      STOP
3437 C      CONTINUE
3438 C      IF(ISLICE.EQ.0)GOTO 180
3439 C      DO 170 I=1,ISLICE
3440 C      READ(IUNIT,* )ISLICE(I),(MXNX(J),J=1,ISLICE(I))
3441 C      IF(JTEST.EQ.0)PRINT*,ISLICE(I),(MXNX(J),J=1,ISLICE(I))
3442 C      IF(ISLICE(I).LT.NDSL2)GOTO 155
3443 C      WRITE(ISCR,152)ISLICE(I),NDSL2
3444 C      152 FORMAT(1X,'*** ERROR: ISLICE=',I2,' GREATER THAN NDSL2=',I2)
3445 C      STOP
3446 C      NTOTAL=0
3447 C      DO 160 J=1,ISLICE(I)
3448 C      MSLICE(J,I)=MXNX(J)/1000
3449 C      NSLICE(J,I)=MXNX(J)-1000*(MXNX(J)/1000)
3450 C      IF(J.GT.1)THEN
3451 C      NDIF=IBABS(NSLICE(J,I)-NSLICE(J-1,I))
3452 C      MDIF=IBABS(MSLICE(J,I)-MSLICE(J-1,I))
3453 C      NTOTAL=NTOTAL+1
3454 C      IF(.NOT.(MDIF.EQ.0.AND.NDIF.GT.0).OR.
3455 C      1 (NDIF.EQ.0.AND.MDIF.GT.0).OR.
3456 C      2 (NDIF.EQ.MDIF))THEN
3457 C      WRITE(ISCR,157)I,MDIF,NDIF
3458 C      157 FORMAT(1X,'*** ERROR: IN SLICE NO.=',I2,' MDIF=',I3,
3459 C      1 I3,' AND NDIF=',I3)
3460 C      STOP
3461 C      END IF
3462 C      END IF
3463 C      160 CONTINUE
3464 C      166 CONTINUE
3465 C      170 CONTINUE
3466 C      CELLS FOR LATER GRAPHING
3467 C      180 READ(IUNIT,1001)
3468 C      READ(IUNIT,* ,ERR=1411)IGPH,NSTGPH,IGPHOP
3469 C      IF(JTEST.EQ.1)PRINT*,IGPH,NSTGPH,IGPHOP
3470 C      GOTO 1420
3471 C      1411 WRITE(ISCR,1412)
3472 C      1412 FORMAT(5X,'*** ERROR READING IGPH, NGPH ***')
3473 C      1420 IF(IGPH.LE.0)GOTO 191
3474 C      IF(IGPH.LE.NDGPB)GOTO 1430
3475 C      WRITE(ISCR,1424)IGPH,NGPH
3476 C      1424 FORMAT(1X,'*** ERROR: IGPH=',I2,' GREATER THAN NDGPB=',I2)
3477 C      STOP
3478 C      1430 JGPH=MINO(IGPH,36)
3479 C      DO 190 I=1,IGPH
3480 C      IF(I.GT.JGPH)READ(IUNIT,* )X
3481 C      IF(I.LE.JGPH)READ(IUNIT,* ,ERR=185)LGPH(I),MGPH(I),NGPH(I),ITYP(I)
3482 C      IF(JTEST.EQ.1)PRINT*,LGPH(I),MGPH(I),NGPH(I),ITYP(I)
3483 C      GOTO 191
3484 C      185 WRITE(ISCR,186)
3485 C      186 FORMAT(5X,'*** (RDCON) ERROR READING LGPH, MGPH, ETC ***')
3486 C      190 CONTINUE
3487 C      191 CONTINUE
3488 C      IGPB=JGPH
3489 C      RETURN
3490 C
3491 C
3492 C-----+
3493 C      SUBROUTINE RDCON2(JTEST,IVER)
3494 C
3495 C      JUNE 1994 K. HESS CE0B HP9000
3496 C      PURPOSE - TO READ IN THE ENVIRONMENTAL DATA AS SEPARATE FILES
3497 C      INCLUDE 'COMM20.FOR'
3498 C      CHARACTER*40 FDATA
3499 C
3500 C      IUNIT=LUCON

```

```

3501      IU=LUCON
3502 C       READ RUN START DATE
3503 C       IF(JTEST.EQ.1)WRITE(ISCR,90)
3504 90      FORMAT(1X,'TIME VARIABLE INPUTS')
3505      READ(IUNIT,1002)
3506 1001 FORMAT(1X)
3507 1002 FORMAT(1X,1X)
3508      READ(IUNIT,*),ERR=95 )IYEAR,MONTH,IDAY,IHOUR,IMIN
3509      IF(IYEAR.LT.100)IYEAR=IYEAR+1900
3510      YEAR=IYEAR
3511      YEAR0=YEAR
3512      YEAR1=YEAR
3513      IF(JTEST.EQ.1)PRINT*, 'YEAR=',IYEAR,MONTH, IDAY,IHOUR,IMIN
3514      GOTO 98
3515 95      WRITE(ISCR,96 )
3516 96      FORMAT(5X,'*** (RDCON) ERROR READING IYEAR, ETC. ***')
3517      STOP
3518 98      CALL JULIAN(IYEAR,MONTH, IDAY,UT,NDAYMO)
3519      UTO=UT+(FLOAT(IHOUR)+FLOAT(IMIN)/60.)/24.
3520      IF(JTEST.EQ.1)PRINT*, 'UTO=',UTO
3521 C       READ TIDAL WATER LEVEL DATA
3522 C       WRITE(ISCR,*) '
3523 C       READ(IUNIT,1001)
3524 C       WRITE(ISCR,*) 'WATER LEVEL DATA'
3525      CALL IRR(IU,ISCR,LUTID,IENDTD,DTID,YTID,NSIGT,NDTID2,TDEVL)
3526 C       READ WIND DATA:
3527 C       WRITE(ISCR,*) '
3528 C       WRITE(ISCR,*) 'WIND DATA'
3529      READ(IUNIT,1001)
3530      READ(IUNIT,*),NSIGW
3531      IENDWN=0
3532      IF(NSIGW.EQ.0)THEN
3533      WRITE(ISCR,*) '
3534      WRITE(ISCR,*) ' NO INPUT DATA '
3535      IENDWN=1
3536      GOTO 300
3537      ENDIF
3538      READ(IUNIT,100)FDATA
3539 100      FORMAT(A40)
3540      WRITE(ISCR,130)FDATA,NSIGW,LUWND
3541 130      FORMAT(5X,'FILE NAME=',A40,/,5X,'NSIG=',I2,' LUT=',I2)
3542      CALL FOPEN(LUWND,FDATA)
3543      DO 220 I=1,2
3544 220      CALL RDWIND(I,IEND)
3545      IENDWN=IEND
3546 300      CONTINUE
3547 C       READ IN RIVER FLOWRATE DATA
3548 C       WRITE(ISCR,*) '
3549 C       WRITE(ISCR,*) 'RIVER FLOWRATE DATA'
3550      READ(IUNIT,1001)
3551      READ(IUNIT,*),NSIGR,NDRIV2,QRIV)
3552      CALL IRR(IU,ISCR,LURIV,IENDRV,DRV1,YRIV,NSIGR,NDRIV2,QRIV)
3553      READ IN OCEANIC SALINITY CONCENTRATION
3554 C       WRITE(ISCR,*) '
3555 C       WRITE(ISCR,*) 'OCEAN SALINITY DATA'
3556      READ(IUNIT,1001)
3557      CALL IRR(IU,ISCR,LUSAL,IENDSO,DSAL,YSAL,NSIGN,NDOCN2,SALOCN)
3558      READ IN OCEANIC TEMPERATURES
3559 C       WRITE(ISCR,*) '
3560 C       WRITE(ISCR,*) 'OCEAN TEMPERATURE DATA'
3561      READ(IUNIT,1001)
3562      CALL IRR(IU,ISCR,LUOCT,IENDTO,DOTP,YOTP,NSIGTO,NDOCN2,TMPOCN)
3563 C       READ IN RIVER TEMPERATURES
3564 C       WRITE(ISCR,*) '
3565 C       WRITE(ISCR,*) 'RIVER TEMPERATURE DATA'
3566      READ(IUNIT,1001)
3567      CALL IRR(IU,ISCR,LURVT,IENDRT,DRV1,YRVT,NSIGRT,NDRIV2,TRIV)
3568      READ IN ADDITIONAL MET DATA
3569 C       WRITE(ISCR,*) '
3570 C       WRITE(ISCR,*) 'ADDITIONAL MET DATA'
3571      READ(IUNIT,1001)
3572      CALL IRR(IU,ISCR,LUMET,IENDMT,DMET,YMET,NSIGM,NDMET2,VMET)
3573 C       READ IN INITIAL CONDITIONS INDEX AND FILE
3574 C       READ(IUNIT,1001)
3575 C       READ(IUNIT,*),ICS
3576      WRITE(ISCR,510)ICS
3577 C       WRITE(ICS,510)ICS
3578 510      FORMAT(1X,'INITIALIZATION. ICS=',I2)
3579      IF(ICS.EQ.0)GOTO 600
3580      READ(IUNIT,100)FINIT
3581      WRITE(ISCR,520)FINIT
3582 520      FORMAT(1X,' INITIALIZATION FILE NAME=',A40)
3583 600      CONTINUE
3584      RETURN
3585      END
3586
3587 C-----SUBROUTINE RDCON3(JTEST,IVER)
3588 C       OCTOBER 1984 K. HESS MEAD VAX 11/70 (REV 8-89)
3589 C       PURPOSE - TO READ IN THE CONTROL FILE
3590 C       INCLUDE 'COMM20.FOR'
3591 C       DIMENSION MNXN(NPMSIZ)
3592 C       IUNIT=LUCON
3593 C       READ OUTPUT FILE NAMES
3594      READ(IUNIT,1001)
3595 1001 FORMAT(1X)
3596      READ(IUNIT,440)FPRINT
3597      READ(IUNIT,440)FGRAPH
3598      READ(IUNIT,440)FMED
3599      WRITE(6,*),'OUTPUT FILE NAMES'
3600      WRITE( 6,440)FPRINT
3601      WRITE( 6,440)FGRAPH
3602      WRITE( 6,440)FMED
3603      FORMAT(1X,A40)
3604      READ(IUNIT,1001)
3605 440      FORMAT(10X,A40)
3606 C       READ END OF DATA STATEMENT
3607      READ(IUNIT,1001)
3608      IF(JTEST.EQ.1)WRITE(ISCR,1303)
3609 1303 FORMAT(1X,'END OF CONTROL FILE READ')
3610      RETURN
3611      END
3612 C-----SUBROUTINE JULIAN(IY,MO,IDAY,UT)
3613 C       JAN 1995 HESS CEOB SGI 4D
3614 C       purpose - to convert year, month, etc to a Julian date and
3615 C       number of hours from start of year
3616 C       NOTE: UT IS NOS CONVENTION, SO NOON ON JANUARY 1
3617 C       IS UT=1.50
3618 C       DIMENSION NMON(12,2)
3619 C       DATA NMON/0,,31,59,30,120,151,181,212,243,273,304,334,
3620 C       1 ,31,60,91,121,152,182,213,244,274,305,335/
3621 C       CONVERT TO 4-DIGIT IF NOT
3622 C       IYEAR=IYRA(IY)
3623 C       look for leap year
3624 C-----FUNCTION IYR4(IY)
3625 C-----SUBROUTINE FFOPEN(IUNIT,FNAME)
3626 C       PURPOSE - TO OPEN A FORMATTED FILE
3627 C-----CHARACTER FNAM*(*)
3628 C-----OPEN(UNIT=IUNIT,FILE=FNAME,FORM='FORMATTED',STATUS='UNKNOWN',
3629 C       1 IOSTAT=IER,ERR=110)
3630      GOTO 140
3631      RETURN
3632      END
3633 C-----SUBROUTINE FUOPEN(IUNIT,FNAME)
3634 C       PURPOSE - TO OPEN A FORMATTED FILE
3635 C-----CHARACTER FNAM*(*)
3636 C-----OPEN(UNIT=IUNIT,FILE=FNAME,FORM='UNFORMATTED',STATUS='UNKNOWN',
3637 C       1 IOSTAT=IER,ERR=110)
3638      GOTO 140
3639      END
3640 C-----SUBROUTINE RDGEO(IGEO)
3641 C       OCTOBER 1984 K. W. HESS ASIO/MEDD VAX 11/750
3642 C       PURPOSE - TO READ IN PARAMETERS, PLUS IFIELD, DEPTH, AND
3643 C       FLAG DATA.
3644 C-----VARIABLES
3645 C       IAH = INDEX TO READ HORIZ, EDDY VISCOSITY: 0=NO
3646 C       IGRID = INDEX TO READ STRETCHED GRIDS:0=NO, 1=READ XE,YE
3647 C       2=READ NSEGX, ETC.
3648 C       NUMBXY = NUMBER OF CHANNEL GRIDS
3649 C       NO, NR = NUMBER OF OCEAN, RIVER BOUNDARY CELLS AS
3650 C       DETERMINED BY IFIELD
3651 C       NO1, NR1 = NUMBER OF OCEAN, RIVER BOUNDARY CELLS AS
3652 C       DETERMINED BY NUMOBC, NUMIV
3653 C       XC,(,) = DISTANCE FROM ORIGIN TO GRID CENTER
3654 C       XL,(,) = DISTANCE TO LOWER SIDE OF GRID M
3655 C-----INCLUDE 'COMM20.FOR'
3656 C-----DIMENSION NUM(NPMSIZ),XL(NPMSIZ),YL(NPMSIZ)
3657 C-----CHARACTER*10 FORM,GTITLE(6)
3658 C-----DATA NDX,NDY,MSIZE,NSIZE/
3659 C-----THIS IS VERSION 2 (8-89)
3660 C-----JVER=2
3661 C-----READ IN MECCA FILE HEADER BLOCK
3662 200      READ(IGEO,202)(GTITLE(N),N=1,6)
3663 202      FORMAT(1X,6A10)
3664      READ(IGEO,J,I) J, ITEST
3665      IF(ITEST.EQ.1)WRITE(ISCR,2200)(GTITLE(N),N=1,6),J
3666 220      FORMAT(1X,'GEOGRAPHY FILE',/1X,'TITLE=',6A10,' J=',I3)
3667      IF(J.NE.JVER)THEN
3668      WRITE(ISCR,204)J,JVER
3669 204      FORMAT(1X,'INPUT FILE IS VERSION=',I2,' RDGEO IS VERSION=',I2,
3670      1 ' RUN IS STOPPED')
3671      STOP
3672      END IF
3673 C-----GRID PARAMETERS
3674 205      READ(IGEO,1002)
3675 1001 FORMAT(1X)
3676 1002 FORMAT(1X,1X)
3677      READ(IGEO,*),ERR=222)NMAX,MMAX_DL
3678      IF(ITEST.EQ.1)WRITE(ISCR,2220)NMAX,MMAX_DL
3679 2220 FORMAT(1X,'NMAX,MMAX_DL=',2I5,E10.5)
3680      GOTO 226
3681 C-----FORMAT(1X,*)ERR=224) (RDGEO) ERROR READING NMAX,MMAX_DL ***
3682 226      IF(IGEO.LE.NDX.AND.NMAX.LE.NDY)GOTO 228
3683 227      WRITE(ISCR,227)NMAX,NDY,MMAX,NDX
3684      FORMAT(1X,*)ERR=227)NMAX,NDY,MMAX,NDX
3685 228      1 GT NDX=,I3)
3686      STOP
3687 229      CONTINUE
3688 230      READ(IGEO,1001)
3689 231      READ(IGEO,*),ERR=232)NCOR,MCOR,BSNLAT,BSNLON,BSNANG
3690      IF(ITEST.EQ.1)WRITE(ISCR,2260)NCOR,MCOR,BSNLAT,BSNLON,BSNANG
3691 2260 FORMAT(1X,'NCOR=',2I5,3F10.3)
3692 232      GOTO 234
3693 233      WRITE(ISCR,233)
3694 234      FORMAT(1X,*)ERR=233) RDGEO. PROBLEM READING LINE NCOR,MCOR, ETC ***
3695      RETURN
3696 235      C-----OCEAN BOUNDARY CONDITIONS
3697 236      READ(IGEO,1001)
3698 237      READ(IGEO,*),NUMOBC
3699 238      IF(IGEO.LE.NDX.AND.NMAX.LE.NDY)GOTO 240
3700 239      WRITE(6,*),'NUMOBC=,I3)
3701 240      DO 241 I=1,NUMOBC
3702 241      READ(IGEO,*),MB1(I),MB2(I),NB1(I),NB2(I),ITPO(I),JTP0(I),ISET1(I),
3703 242      1 ISET2(I)
3704 243      READ(IGEO,*),MB1(I),MB2(I),NB1(I),NB2(I),ITPO(I),ISET1(I),ISET2(I)
3705 244      IF(ISET2.EQ.1)WRITE(ISCR,2360)MB1(I),MB2(I),NB1(I),NB2(I),ISET1(I)
3706 245      1 ,ISET2(I)
3707 246      1 ISET2(I)
3708 247      1 ,ISET2(I)
3709 248      1 ,ISET2(I)
3710 249      1 ,ISET2(I)
3711 250      1 ,ISET2(I)
3712 251      1 ,ISET2(I)
3713 252      1 ,ISET2(I)
3714 253      1 ,ISET2(I)
3715 254      1 ,ISET2(I)
3716 255      1 ,ISET2(I)
3717 256      1 ,ISET2(I)
3718 257      1 ,ISET2(I)
3719 258      1 ,ISET2(I)
3720 259      1 ,ISET2(I)
3721 260      1 ,ISET2(I)
3722 261      1 ,ISET2(I)
3723 262      1 ,ISET2(I)
3724 263      1 ,ISET2(I)
3725 264      1 ,ISET2(I)
3726 265      1 ,ISET2(I)
3727 266      1 ,ISET2(I)
3728 267      1 ,ISET2(I)
3729 268      1 ,ISET2(I)
3730 269      1 ,ISET2(I)
3731 270      1 ,ISET2(I)
3732 271      1 ,ISET2(I)
3733 272      1 ,ISET2(I)
3734 273      1 ,ISET2(I)
3735 274      1 ,ISET2(I)
3736 275      1 ,ISET2(I)
3737 276      1 ,ISET2(I)
3738 277      1 ,ISET2(I)
3739 278      1 ,ISET2(I)
3740 279      1 ,ISET2(I)
3741 280      1 ,ISET2(I)
3742 281      1 ,ISET2(I)
3743 282      1 ,ISET2(I)
3744 283      1 ,ISET2(I)
3745 284      1 ,ISET2(I)
3746 285      1 ,ISET2(I)
3747 286      1 ,ISET2(I)
3748 287      1 ,ISET2(I)
3749 288      1 ,ISET2(I)
3750 289      1 ,ISET2(I)
3751 290      1 ,ISET2(I)
3752 291      1 ,ISET2(I)

```

```

3753      STOP
3754      END IF
3755  240 CONTINUE
3756 C----- READ IN RIVER BOUNDARIES
3757  242 READ(IGEO,1001)
3758      READ(IGEO,*) NUMRIV
3759      IF(ITEST,EQ.1)WRITE(ISCR,2450)NUMRIV
3760  2450 FORMAT(1X,'NUMRIV',I5)
3761      IF(NUMRIV,LE,0)GOTO 248
3762      DO 246 I=1,NUMRIV
3763  246 READ(IGEO,*)M1(I),M2(I),NR1(I),NR2(I),JTPR(I),ISETR(I)
3764      IF(ITEST,EQ.1)WRITE(ISCR,2460)M1(I),M2(I),NR1(I),NR2(I),JTPR(I)
3765  2460 FORMAT(1X,'M1=',I15)
3766      JTPR(I)=MAX0(1,MIN0(2,JTPR(I)))
3767      IF(.NOT.(M1(I).LE.M2(I).AND.NR1(I).LE.NR2(I)))THEN
3768      WRITE(ISCR,1238)I,M1(I),M2(I),NR1(I),NR2(I)
3769  1238 FORMAT(1X,'*** ERROR *** AT RIVER BOUNDARY # ',I2,' M1=',I3,
3770      ' IS GT M2=',I3,', ' OR NB1=',I3,' IS GT NB2=',I3)
3771      STOP
3772      END IF
3773      IF(.NOT.(M1(I).EQ.M2(I).OR.NR1(I).EQ.NR2(I)))THEN
3774      WRITE(ISCR,1239)I,M1(I),M2(I),NR1(I),NR2(I)
3775  1239 FORMAT(1X,'*** ERROR *** AT RIVER BOUNDARY # ',I2,' M1=',I3,
3776      ' IS .NE. M2=',I3,', ' OR NB1=',I3,' IS .NE. NB2=',I3)
3777      STOP
3778      END IF
3779  246 CONTINUE
3780 C----- READ THE CELL STATUS FIELD (IFIELD)
3781  248 READ(IGEO,1001)
3782      READ(IGEO,2490,ERR=249)FORM
3783  2490 FORMAT(1X,A10)
3784      IF(ITEST,EQ.1)WRITE(ISCR,2491)FORM
3785  2491 FORMAT(1X,'CELL STATUS: FORM=',A10)
3786      GOTO 2494
3787  2494 WRITE(ISCR,2492)
3788  2492 FORMAT(1X,'*** ERROR READING FORM ***')
3789  2494 READ(IGEO,*)NPERL,KOCNRC
3790      IF(ITEST,EQ.1)WRITE(ISCR,2496)NPERL,KOCNRC
3791  2496 FORMAT(1X,'NPERL=',I5)
3792      IF(KOCNRC.GE.4.AND.KOCNRC.LE.8)GOTO 2502
3793      WRITE(ISCR,2501)KOCNRC
3794  2501 FORMAT(1X,'*** ERROR: KOCNRC=',I3,' NOT BETWEEN 4 AND 8')
3795      STOP
3796  2502 IF(ITEST,EQ.1)WRITE(ISCR,*) IFIELD'
3797      KMAX=1*(NMAX-1)/NPERL
3798      DO 250 K=1,KMAX
3799      N1=1+(K-1)*NPERL
3800      N2=MIN0(N1+NPERL-1,NMAX)
3801      DO 250 M=1,MAX
3802      READ(IGEO,FORM)(IFIELD(N,M),N=N1,N2)
3803      IF(ITEST,EQ.1)WRITE(ISCR,2497)(IFIELD(N,M),N=N1,N2)
3804  2497 FORMAT(40I2)
3805  250 CONTINUE
3806 C----- READ THE DEPTHS
3807      READ(IGEO,1001)
3808      READ(IGEO,2490)FORM
3809      IF(ITEST,EQ.1)WRITE(ISCR,2642)FORM
3810  2642 FORMAT(1X,'DEPTHS: FORM=',A10)
3811      READ(IGEO,*,ERR=264)NCOL2,HMSL,CON2M
3812      IF(ITEST,EQ.1)WRITE(ISCR,2643)NCOL2,HMSL,CON2M
3813  2643 FORMAT(1X,'NCOL2=',I5,2F10.3)
3814      GOTO 266
3815  266 WRITE(ISCR,265)
3816  265 FORMAT(1X,'*** ERROR READING NCOL2,HMSL,CON2M ***')
3817  265 FORMAT(1X,'*** ERROR READING NCOL2,HMSL,CON2M ***')
3818  266 NSWEEP=1*(NMAX-1)/NCOL2
3819      DO 285 NN=1,NSWEEP
3820      N1=1+NCOL2*(NN-1)
3821      N2=MIN0(NMAX,N1+NCOL2-1)
3822      DO 280 M=1,MAX
3823      READ(IGEO,FORM)(NUM(N),N=N1,N2)
3824      IF(ITEST,EQ.1)WRITE(ISCR,264)NUM(N),N=N1,N2)
3825      CONVERT DEPTHS TO METERS AND ADD HMSL
3826      DO 275 N=N1,N2
3827      D(N,M)=0.0
3828  275 IF(IFIELD(N,M).GT.0)D(N,M)=FLOAT(NUM(N))*CON2M+HMSL
3829  280 CONTINUE
3830  285 CONTINUE
3831 C----- SET GRID WIDTHS AND AREA
3832      READ(IGEO,1001)
3833      READ(IGEO,*,ERR=301)NUMBXV
3834  301 FORMAT(1X,'WIDTH DATA: NUMBXV',I5)
3835  302 FORMAT(1X,'*** ERROR READING NUMBXV ***')
3836  302 FORMAT(1X,'*** ERROR READING NUMBXV ***')
3837  305 CONTINUE
3838 C----- GRID X,Y COORDINATES
3839      DO 311 I=1,NPMSIZ
3840      XL(I)=I-1
3841  311 YL(I)=I-1
3842      DO 320 M=1,MAX
3843      DO 320 N=1,NMAX
3844      BX(N,M)=1.0
3845      BY(N,M)=1.0
3846      AREA(N,M)=(XL(M+1)-XL(M))*(YL(N+1)-YL(N))
3847      IF(IFIELD(N,M)/10.EQ.1)AREA(N,M)=0.5*AREA(N,M)
3848  320 CONTINUE
3849 C----- READ THRU THE CHANNEL WIDTHS
3850      IF(NUMBXV,LE,0)GOTO 350
3851      DO 340 J=1,NUMBXV
3852      READ(IGEO,*)M,N,BX(N,M),BY(N,M),F1
3853      IF(ITEST,EQ.1)WRITE(ISCR,311)J,M,N,BX(N,M),BY(N,M),F1
3854  311 FORMAT(1X,'J=',I4,' M,',I2,' N,',I2,F10.4)
3855      AREA(M,M)=AREA(N,M)*F1
3856  340 CONTINUE
3857  350 CONTINUE
3858 C----- IF(ITEST,EQ.1)WRITE(ISCR,3391)
3859  3391 FORMAT(1X,'END OF GEOGRAPHY DATA FILE')
3860 C----- CHECK BOUNDARIES
3861  400 CALL CHECKS
3862      RETURN
3863  END
3864 C----- SUBROUTINE CHECKS
3865 C----- OCTOBER 1986 HESS MEAD VAX 780
3866 C----- PURPOSE - TO CHECK FOR CONSISTANCY BETWEEN RIVER AND OCEAN
3867 C----- BOUNDARIES AND THE IFIELD SPECIFICATION

```

```

4005 C IF(IFIELD(N,M+1).LT.10.OR.MOD(II,10).EQ.1.OR.MOD(II,10).EQ.3)IXP=0
4006 C Y FLOW
4007 C IYM=1
4008 C IF(N.EQ.1)THEN
4009 C IYM=0
4010 C ELSE
4011 C II=IFIELD(N-1,M)
4012 C IF(II.LT.10.OR.MOD(II,10).EQ.2.OR.MOD(II,10).EQ.3)IYM=0
4013 C END IF
4014 C IYP=1
4015 C II=IFIELD(N,M)
4016 C IF(IFIELD(N+1,M).LT.10.OR.MOD(II,10).EQ.2.OR.MOD(II,10).EQ.3)IYP=0
4017 C     CHECK FOR ADJACENTS
4018 C IFLows(1)=IXP
4019 C IFLows(2)=IYP
4020 C IFLows(3)=IXM
4021 C IFLows(4)=IYM
4022 C IFLows(5)=IFLows(1)
4023 C IFLows(6)=IFLows(2)
4024 C DO 460 I=1,4
4025 C IF(IFLows(I)+IFLows(I+1).EQ.0)GOTO 480
4026 C 460    CONTINUE
4027 C WRITE(ISCR,470)N,M,IXP,IYP,IXM,IYM
4028 C 470    FORMAT(1X,*** WARNING, TRIANGULAR CELL AT N='I3,', M='I3,
4029 C 1' HAS INCORRECT SIDES/,1X,'IXP,IYP,IXM,IYM=',4I4)
4030 C NBAD=NBAD+1
4031 C 480    CONTINUE
4032 C     CHECK FOR TOTAL TRIANGULAR CELLS NOT WELL DEFINED
4033 C IF(NBAD.GT.0)THEN
4034 C WRITE(ISCR,495)NBAD
4035 C 495    FORMAT(1X,***ERROR: TRIANGULAR CELLS NOT WELL DEFINED=',I4)
4036 C STOP
4037 C END IF
4038 C 500    CONTINUE
4039 C     CHECK FOR CORRESPONDANCE BETWEEN (1) BOUNDARIES AND (2) INPUT SIG
4040 C WRITE(ISCR,' '
4042 C WRITE(ISCR,*)'CHECK BOUNDARIES AND DATA'
4043 C NBAD=0
4044 C     ocean boundaries
4045 C IF(NUMOBC.GT.0)THEN
4046 C IF(NSIGTD.EQ.0)THEN
4047 C     WRITE(ISCR,*)'**NO WATER LEVEL SIGNALS AT OCEAN BOUNDARY**'
4048 C NBAD=1
4049 C ENDIF
4050 C IF((KONCEN.EQ.1.OR.KONCEN.EQ.3).AND.NSIGS.EQ.0)THEN
4051 C     WRITE(ISCR,*)'**NO SALINITY SIGNALS AT OCEAN BOUNDARY**'
4052 C NBAD=1
4053 C ENDIF
4054 C IF((KONCEN.EQ.2.OR.KONCEN.EQ.3).AND.NSIGT.EQ.0)THEN
4055 C     WRITE(ISCR,*)'**NO TEMPERATURE SIGNALS AT OCEAN BOUNDARY**'
4056 C NBAD=1
4057 C ENDIF
4058 C ENDIF
4059 C     river boundaries
4060 C IF(NUMRIV.GT.0)THEN
4061 C IF(NSIGR.EQ.0)THEN
4062 C     WRITE(ISCR,*)'**NO FLOW RATE DATA AT RIVER BOUNDARY**'
4063 C NBAD=1
4064 C ENDIF
4065 C IF((KONCEN.EQ.2.OR.KONCEN.EQ.3).AND.NSIGT.EQ.0)THEN
4066 C     WRITE(ISCR,*)'**NO TEMPERATURE SIGNALS AT RIVER BOUNDARY**'
4067 C NBAD=1
4068 C ENDIF
4069 C ENDIF
4070 C     air-sea boundaries
4071 C IF((KONCEN.EQ.2.OR.KONCEN.EQ.3).AND.NSIGW.EQ.0)THEN
4072 C     WRITE(ISCR,*)'**NO WIND DATA AT AIR-SEA BOUNDARY**'
4073 C NBAD=1
4074 C ENDIF
4075 C IF((KONCEN.EQ.2.OR.KONCEN.EQ.3).AND.NSIGM.EQ.0)THEN
4076 C     WRITE(ISCR,*)'**NO ADDITIONAL MET DATA AT AIR-SEA BOUNDARY**'
4077 C NBAD=1
4078 C ENDIF
4079 C IF(NBAD.EQ.1)STOP
4080 C RETURN
4081 C END
4082 C
4083 C -----
4084 C -----
4085 C
4086 C SUBROUTINE RDICS
4087 C     FEBRUARY 1997      K. W. HESS      IRIS
4088 C     PURPOSE - TO READ IN THE INITIAL CONDITIONS
4089 C     INCLUDE 'COMM20.FOR'
4090 C     READ THE INDICES
4091 C     LUICS40C
4092 C     WRITE(ISCR,100)LUICS
4093 C 100    FORMAT(1X,'NOW READING THE ICS FILE ON UNIT=',I2)
4094 C     CALL FOPEN(LUICS,FINIT)
4095 C     READ(LUICS)NIX,MIX,LBOT1,NSTET,UT01,YEAR01,K
4096 C     WRITE(ISCR,110)UT1,YEAR1,NSTET,K
4097 C 110    FORMAT(5X,'UT1=','F10.4,' ,YEAR1=','F6.1,' ,NSTET=','I6,' ,K=',I2)
4098 C     READ(LUICS)SE,UE,VE,SOLD,UHOLD,VHOLD,AH,AV,PHI,TBX,TBY,
4099 C 1,U,V,W,THETA1,THETA2,THETA3
4100 C     IF(K.GE.10)READ(LUICS)AH3,WX,WY,TSX,TSY
4101 C     IF(MOD(K,10).GT.0)READ(LUICS)S,T,DV,BI,NSTINF
4102 C     RESET SECONDARY VARIABLES
4103 C     YEAR1=YEAR01
4104 C     UT1=UT01
4105 C     DO 120 M=1,MMAX
4106 C     MP=MINO(M+1,MMAX)
4107 C     DO 120 N=1,NMAX
4108 C     IF(IFIELD(N,M).LT.10)GOTO 120
4109 C     NP=MINO(N+1,NMAX)
4110 C     UH(N,M)=UE(N,M)* .5*(D(N,M)+SE(N,M)+D(N,MP)+SE(N,MP)+E)*BX(N,M)
4111 C     VH(N,M)=VE(N,M)* .5*(D(N,M)+SE(N,M)+D(NP,MP)+SE(NP,MP)+E)*BY(N,M)
4112 C     AH(N,M)= .25*(AH(N,M)+AH(N,MP)+AH(NP,MP))
4113 C 120    CONTINUE
4114 C     IF(INTER.GT.0.AND.KONC.LT.10)THEN
4115 C     DO 140 M=1,MMAX
4116 C     DO 140 N=1,NMAX
4117 C     DO 140 L=1,LBOT
4118 C     DO 140 I=1,IHM
4119 C     AH3(L,N,M)=AH(N,M)
4120 C     ENDIF
4121 C     WRITE(ISCR,*)'COMPLETE RDICS'
4122 C     CLOSE(LUICS)
4123 C     RETURN
4124 C END
4125 C
4126 C -----
4127 C
4128 C SUBROUTINE ZEROS
4129 C     APRIL 1988      K. W. HESS
4130 C     PURPOSE - INITIALIZE THE PARAMETERS TO ZERO BEFORE
4131 C           INPUT FILES HAVE BEEN READ IN.
4132 C           VARIABLES -
4133 C           INCLUDE 'COMM20.FOR'
4134 C           INITIALIZE VISCOSITIES AND DEPTHS
4135 C           LBOT=LSIZE
4136 C           DO 130 M=1,MSIZE
4137 C           DO 130 N=1,NSIZE
4138 C           AREA(N,M)=0.0
4139 C           TSX(N,M)=0.0
4140 C           TSY(N,M)=0.0
4141 C           WX(N,M)=0.0
4142 C           WY(N,M)=0.0
4143 C           TBX(N,M)=0.0
4144 C           TBY(N,M)=0.0
4145 C           UH(N,M)=0.0
4146 C           VH(N,M)=0.0
4147 C           SE(N,M)=0.0
4148 C           SEP(N,M)=0.0
4149 C           SEPP(N,M)=0.0
4150 C           UHP(N,M)=0.0
4151 C           VHP(N,M)=0.0
4152 C           SOLD(N,M)=0.0
4153 C           UHOLD(N,M)=0.0
4154 C           VHOLD(N,M)=0.0
4155 C           THETA1(N,M)=0.0
4156 C           THETA2(N,M)=0.0
4157 C           THETA3(N,M)=0.0
4158 C           UE(N,M)=0.0
4159 C           VE(N,M)=0.0
4160 C           AH(N,M)=0.0
4161 C           AHC(N,M)=0.0
4162 C           FEDGE(N,M)=1.0
4163 C           PHI(N,M)=0.0
4164 C           MFLUX(N,M)=0.0
4165 C           NFLUX(N,M)=0.0
4166 C           LOOP OVER LEVELS
4167 C           DO 120 L=1,LSIZE
4168 C           U(L,N,M)=0.0
4169 C           V(L,N,M)=0.0
4170 C           W(L,N,M)=0.0
4171 C           AH3(L,N,M)=0.0
4172 C           S(L,N,M)=0.0
4173 C           T(L,N,M)=0.0
4174 C           RI(L,N,M)=0.0
4175 C           DV(L,N,M)=0.0
4176 C 120    AV(L,N,M)=0.0
4177 C 130    CONTINUE
4178 C
4179 C           DO 135 L=1,LSIZE
4180 C           DO 135 I=1,LM2SIZ
4181 C 135    NSTINF(L,I)=0
4182 C           SET HEAT FLUXES
4183 C           QI=0.0
4184 C           QA=0.0
4185 C           QB=0.0
4186 C           QS=0.0
4187 C           QE=0.0
4188 C           RETURN
4189 C
4190 C -----
4191 C           SUBROUTINE YTIM8(YT8,UT,YEAR)
4192 C           create a date from year and Julian day
4193 C           REAL*8 YT8,UT,DAYS
4194 C           look for leap year
4195 C           IYEAR=YEAR
4196 C           DAYS=365
4197 C           IF(MOD(IYEAR,4).EQ.0.AND.(.NOT.(MOD(IYEAR,100).EQ.0.AND.
4198 C 1 MOD(IYEAR,400).NE.0)))DAYS=366.
4199 C           UT8=UT
4200 C           YT8=FLOAT(IYEAR-1900)+(UT8-1.)/DAYS
4201 C           RETURN
4202 C           END
4203 C
4204 C -----
4205 C
4206 C           SUBROUTINE RR(YT,ISCR,LUT,IEND,DD,YD,NSIG,NN,VALS,FINAL)
4207 C           generic read for input data records
4208 C           DIMENSION DD(2),YD(2),VALS(2,NN),FINAL(NN)
4209 C           REAL*8 YT1,YT2
4210 C           DO 90 N=1,NN
4211 C 90    FINAL(N)=0.0
4212 C           IF(NSIG.EQ.0.OR.IEND.EQ.1)RETURN
4213 C           CALL YTIM8(YT1,DD(1),YD(1))
4214 C           CALL YTIM8(YT2,DD(2),YD(2))
4215 C           IF(YT.LT.YT1)THEN
4216 C             WRITE(ISCR,95)LUT,YT,YT1
4217 C             FORMAT(1X,*' EARLIER THAN FIRST DATA TIME ON UNIT=',I2,/,1
4218 C 1 YT1=YT1,F12.8,' YT1=',F12.8,' **')
4219 C             RETURN
4220 C           ENDIF
4221 C 100    IF(YT.GT.YT2)THEN
4222 C             DO 120 N=1,NSIG
4223 C 120    VALS(1,N)=VALS(2,N)
4224 C             DD(1)=DD(2)
4225 C             YD(1)=YD(2)
4226 C             YT1=YT2
4227 C             READ(LUT,* END=130)YD(2),DD(2),(VALS(2,N),N=1,NSIG)
4228 C             CALL YTIM8(YT2,DD(2),YD(2))
4229 C             GOTO 100
4230 C           ENDIF
4231 C             GOTO 150
4232 C 130    IEND=1
4233 C             WRITE(ISCR,140)LUT
4234 C 140    FORMAT(1X,*' NO MORE DATA ON UNIT=',I2,' **')
4235 C             RETURN
4236 C           END
4237 C 150    F2=(YT-YT1)/(YT2-YT1)
4238 C             DO 160 N=1,NSIG
4239 C 160    FINAL(N)=(1.-F2)*VALS(1,N)+F2*VALS(2,N)
4240 C             RETURN
4241 C           END
4242 C
4243 C -----
4244 C
4245 C           SUBROUTINE IRR(IUNIT,ISCR,LUT,IEND,DD,YD,NSIG,NN,VALS)
4246 C           DIMENSION DD(2),YD(2),VALS(2,NN)
4247 C           CHARACTER*40 FDATA
4248 C           INITIAL READ OF DATA
4249 C           IEND=0
4250 C           READ(IUNIT,100)
4251 C 100    FORMAT(1X)
4252 C           READ(IUNIT,*)NSIG
4253 C           IF(NSIG.GT.0)GOTO 110
4254 C           WRITE(ISCR,*)' NO INPUT DATA '
4255 C           GOTO 140
4256 C           read file

```

```

4257 110 READ(IUNIT,120)FDATA
4258 120 FORMAT(A40)
4259 WRITE(130,120)FDATA,NSIG,LUT
4260 130 FORMAT(5X,'FILE NAME=',A40,/,5X,'NSIG=',I2,' LUT=',I2)
4261 CALL FOPEN(LUT,FDATA)
4262 READ(LUT,*END=140)YD(1),DD(1),(VALS(1,N),N=1,NSIG)
4263 READ(LUT,*END=140)YD(2),DD(2),(VALS(2,N),N=1,NSIG)
4264 IEND=0
4265 GOTO 150
4266 140 IEND=1
4267 150 RETURN
4268 END
4269 c
4270 c-----+
4271 c
4272 SUBROUTINE RDWIND(I,IEND)
4273 c PURPOSE - TO READ THE METEOROLOGICAL FILES
4274 c VARIABLES -
4275 c   ITYPE1 = INDEX FOR WIND OR STRESS.
4276 c   1=WIND, 2=STRESS
4277 c   ITYPE2 = INDEX FOR HORIZONTAL ATMOSPHERIC PRESSURE GRADIENT
4278 c   0=None, 1=VALUES TO BE READ
4279 c   ITYPE3 = INDEX FOR COORDINATES FOR WIND/STRESS AND PRESSURE
4280 c   0 IN MODEL COORDS
4281 c   1=IN EAST, NORTH
4282 INCLUDE 'COMM20.FOR'
4283 COMMON/METDX/TMET8(2),ITYPE1
4284 DIMENSION ATX(NSIZE,MSIZE),ATY(NSIZE,MSIZE)
4285 REAL*8 TMET8,UTM
4286 c   set end-of-file index
4287 IEND=0
4288 c   read data
4289 READ(LUWND,END=110)DATE1,DATE2,ITYPE1,ITYPE2,ITYPE3,NIX,MIX
4290 IF(DATE1.LT.JYR.370).THEN
4291 YMET(I)=DATE1
4292 DMET(I)=DATE2
4293 ELSE
4294 YMET(I)=DATE2
4295 DMET(I)=DATE1
4296 ENDIF
4297 c   read arrays
4298 DPADX=0.
4299 DPADY=0.
4300 IF(NIX.GT.1.AND.MIX.GT.1)THEN
4301 IF(ITYPE2.EQ.0)READ(LUWND)((ATX(N,M),N=1,NIX),M=1,MIX),
4302 1((ATY(N,M),N=1,NIX),M=1,MIX)
4303 IF(ITYPE2.EQ.1)READ(LUWND)((ATX(N,M),N=1,NIX),M=1,MIX),
4304 1((ATY(N,M),N=1,NIX),M=1,MIX),DPADX,DPADY
4305 ELSE
4306 c   read single wind speed/stress
4307 IF(ITYPE2.EQ.0)READ(LUWND)AT1,AT2
4308 IF(ITYPE2.EQ.1)READ(LUWND)AT1,AT2,DPADX,DPADY
4309 DO M=1,MSIZE
4310 DO N=1,NSIZE
4311 ATX(N,M)=AT1
4312 ATY(N,M)=AT2
4313 ENDDO
4314 ENDDO
4315 ENDDIF
4316 c   rotate winds to model basin angle
4317 IF(ITYPE3.EQ.1)THEN
4318 ARG=RAD*BSNANG
4319 DO M=1,MSIZE
4320 DO N=1,NSIZE
4321 ATX1=-COS(ARG)*ATY(N,M)-SIN(ARG)*ATX(N,M)
4322 ATY1=COS(ARG)*ATX(N,M)-SIN(ARG)*ATY(N,M)
4323 ATX(N,M)=ATX1
4324 ATY(N,M)=ATY1
4325 ENDDO
4326 ENDDO
4327 DPADX1=-COS(ARG)*DPADY-SIN(ARG)*DPADX
4328 DPADY1=COS(ARG)*DPADX-SIN(ARG)*DPADY
4329 DPADX=DPADX1
4330 DPADY=DPADY1
4331 ENDDIF
4332 c   create date
4333 CALL YTME8(UTM,DMET(I),YMET(I))
4334 TMET8(I)=UTM
4335 WRITE(130,'RDWIND: I, NSTI, ITYPE1, ITYPE2, ITYPE3, DMET(I), YMET(I),
4336 1, TMET8(I), NIX, MIX')
4337 100 FORMAT(5X,'RDWIND: I, NSTI=',2I6,', ITYPE1,2,3=',3I6,/,
4338 1,5X,'DY,YR',2F12.5,' TMET8(I)=',F12.7,/,5X,'NIX,MIX=',2I5)
4339 WRITE(130,105)ATX(1,1),ATY(1,1),DPADX,DPADY
4340 105 FORMAT(5X,'ATX,ATY(1,1)='',ZE12.4,' DPADX,DPADY=''',ZE12.4)
4341 c   save data
4342 DO N=1,NSIZE
4343 DO M=1,MSIZE
4344 FX(I,N,M)=ATX(N,M)
4345 FY(I,N,M)=ATY(N,M)
4346 ENDDO
4347 ENDDO
4348 IEND=N-IEND
4349 RETURN
4350 c   end of data
4351 110 IEND=1
4352 WRITE(6,*)'END OF MET DATA REACHED'
4353 IEND=N-IEND
4354 RETURN
4355 END
4356 c=====+
4357 c   MECCA: ANALYS
4358 c
4359 c-----+
4360 c
4361 SUBROUTINE ANALYS
4362
4363 c JULY 1988 K. HESS MEAD VAX780
4364 c PURPOSE - TO CALL THE ANALYSIS ROUTINES
4365 INCLUDE 'COMM20.FOR'
4366 CALL CHECK2
4367 RETURN
4368 END
4369 c
4370 c-----+
4371 c
4372 SUBROUTINE CHECK2
4373
4374 c JULY 1988 K. HESS MEAD VAX780
4375 c PURPOSE - TO CALL THE ANALYSIS ROUTINES
4376 INCLUDE 'COMM20.FOR'
4377 c   CHECK FOR LARGE WATER LEVEL VALUES
4378 ISTOP=0
4379 SEMAX=3.
4380 DO 110 N=1,NMAX
4381 DO 110 M=1,MMAX
4382 IF(ABS(SE(N,M)).GT.SEMAX)THEN
4383 WRITE(6,105)NSTI,N,M,SE(N,M),D(N,M)
4384 105 FORMAT(1X,'ANALYSIS: NSTI=',I6,' N,M=',2I4,' SE=',F6.2,' D=',F6.2)
4385 ISTOP=1
4386 ENDIF
4387 110 CONTINUE
4388 c   CHECK FOR LARGE SALINITY VALUES
4389 IF(KONCEN.EQ.0)RETURN
4390 IF(IPRNTR1.EQ.0)RETURN
4391 SALMIN= 1.E+10
4392 SALMAX=-1.E+10
4393 SALMAX=S(L,N,M)
4394 DO M=1,MMAX
4395 DO N=1,NMAX
4396 IF(IFIELD(N,M).GT.0)THEN
4397 DO L=1,LBOT
4398 c   check for max
4399 IF(S(L,N,M).GT.SALMAX)THEN
4400 MSMAX=M
4401 NSMAX=N
4402 LSMAX=L
4403 SALMAX=S(L,N,M)
4404 IF(S(L,N,M).LT.SALMIN)THEN
4405 MSMIN=M
4406 NSMIN=N
4407 LSMIN=L
4408 SALMIN=S(L,N,M)
4409 ENDIF
4410 ENDIF
4411 ENDO
4412 ENDIF
4413 ENDO
4414 ENDO
4415 WRITE(6,150)UNT,NSTI,SALMAX,MSMAX,NSMAX,LSMAX
4416 150 FORMAT(1X,'AT UT='',F10.4,' NSTI='',I6,T30,' SALMAX='',F10.2,
4417 1' AT M,N,L='',3I4)
4418 WRITE(6,160)SALMIN,MSMIN,NSMIN,LSMIN
4419 160 FORMAT(T30,' SALMIN='',F10.2,' AT M,N,L='',3I4)
4420
4421 RETURN
4422 END
4423 c=====+
4424 c   MECCA FILE : CONC.FOR - CONCENTRATION SUBROUTINES
4425 c-----+
4426 c
4427 SUBROUTINE CONC2
4428 c MARCH 1986 K. W. HESS (LAST REVISED 23 JULY 87)
4429 c PURPOSE - SUPER-STREAMLINED, SQUARE-GRID VERSION,
4430 c TO COMPUTE NEW DISTRIBUTION OF CONCENTRATE,
4431 c WHICH INCLUDES VARIABLE WIDTH AND VARIABLE HORIZONTAL
4432 c VISCOSITY. NEW FORMULATION OF UPPER BOUNDARY CONDITION
4433 c INITIALIZE BOUNDARY CONCENTRATIONS
4434 INCLUDE 'COMM20.FOR'
4435 COMMON/BNDX4/ISALT,ITEMP,NB
4436 c   CHECK FOR TIME
4437 IF(HR1.LT.HRCON1)RETURN
4438 c   SET INDICES: 1=DO IT, 2=SKIP
4439 ISALT=2
4440 IF(KONCEN.EQ.1.OR.KONCEN.EQ.3)ISALT=1
4441 ITEM=2
4442 IF(KONCEN.EQ.2.OR.KONCEN.EQ.3)ITEMP=1
4443 c
4444 c   GET OCEAN BOUNDARY CONDITIONS
4445 CALL BNDRY3
4446 c   GET RIVER BOUNDARY CONDITIONS
4447 CALL BNDRY4
4448 c   SOLVE GENERALIZED TRANSPORT EQUATION
4449 CALL GFLUX
4450 END
4451 c
4452 c-----+
4453 c
4454 SUBROUTINE BNDRY3
4455 c OCTOBER 1984 K. W. HESS MEAD VAX11/750
4456 c PURPOSE - TO SET THE OCEANIC SALINITY AND TEMPERATURE
4457 c BOUNDARY CONDITIONS
4458 c   VARIABLES -
4459 c     IDIR = INFLOW DIRECTION PARAMETER:1=X DIR, 2=Y DIR
4460 c     ISALT = INFLOW PARAMETER:1=INFLOW IS IN POSITIVE DIR.
4461 c           2=INFLOW IS IN NEGATIVE DIRECTION
4462 c     ISENSE = INFLOW PARAMETER:1=INFLOW IS IN POSITIVE DIRECTION
4463 c     ITO = TYPE OF OCEAN BOUNDARY CONDITION
4464 c       +/-1 : OUTFLOW IN +/-X DIRECTION
4465 c       +/-2 : OUTFLOW IN +/-Y DIRECTION
4466 c     JTP0 = TYPE OF OCEAN BOUNDARY CONDITION
4467 c       1 : WATER LEVEL SPECIFICATION
4468 c       2 : TRANSPORT OUTFLOW
4469 c       3 : ORLANSKI RADIATION OUTFLOW (REQUIRES WL)
4470 c     ITPR = DIRECTION OF RIVER BOUNDARY INFLOW TO ESTUARY
4471 c       +/-1 : INFLOW IN +/-X DIRECTION
4472 c       +/-2 : INFLOW IN +/-Y DIRECTION
4473 c     JTPR = TYPE OF RIVER BOUNDARY CONDITION
4474 c       1 : CHANNEL
4475 c       2 : FALLS
4476 c
4477 c   ORIV = RIVER FLOWRATE (M**3/S)
4478 INCLUDE 'COMM20.FOR'
4479 COMMON/BNDX4/ISALT,ITEMP,NB
4480 DIMENSION SN(LSIZE),TN(LSIZE)
4481 CHARACTER*6 ANAMES(2)
4482 DATA ANAMES/'OUTFL0','INFL0',//,HRINF/6./,IP/0/
4483 c   MAKE SURE THERE ARE BOUNDARIES TO SET
4484 IF(NUMOBC.LE.0)RETURN
4485 c   SET OCEANIC CONDITIONS
4486 CALL BSTATE
4487 NHRINF=HRINF*IHR
4488 CINI=DTI/DL
4489 IP1=IPRNTR1
4490 IF(IP.EQ.0)IP1=0
4491 c   LOOP THRU OCEANIC BOUNDARIES
4492 NB=2
4493 DO 360 IB=1,NUMOBC
4494 IF(IP1.EQ.0)WRITE(130,'OCEAN SALINITY BND: IB='',IB,
4495 1, 'NSTI='',NSTI
4496 MP=MB1(IB)
4497 ML=MB2(IB)
4498 NB=NB1(IB)
4499 NL=NB2(IB)
4500 c   SET DIRECTIONAL PARAMETER, IDIR: 1=X, 2=Y
4501 IDIR=IABS(JTP0(IB))
4502 ISENSE=(3-ISIGN(1,JTP0(IB)))/2
4503 c   OUTFLOW SENSE:FOR +DIRECTION, ISENSE=1; FOR -DIR, ISENSE=2
4504 ND1=(IDIR/2)*(2*ISENSE-3)
4505 MD1=(1-IDIR/2)*(2*ISENSE-3)
4506 c   LOOP THRU EACH GRID
4507 DO 295 N=NF,NL
4508 DO 295 M=MF,ML

```

```

4509      N1=N+ND1
4510      M1=M+MD1
4511      NB=NB+1
4512 C      LOOP OVER DEPTHS
4513      DO 220 L=1,LBOT
4514      GOTO(100, 110) IDIR
4515  100   UL=UE(N,M-2+ISENSE)+U(L,N,M-2+ISENSE)
4516      GOTO 120
4517  110   UL=VE(N-2+ISENSE,M)+V(L,N-2+ISENSE,M)
4518  120   CONTINUE
4519      IF(-UL*ISIGN(1, ISENSE-2).LE.0.0)GOTO 150
4520 C      OUTFLOW CONDITIONS
4521  130   JFLOW=1
4522      NSTINF(L,NB)=NSTIT
4523 C      FIRST-ORDER ADVECTION
4524      F0=ABS(CIN1*UL)
4525      F1=F0*(2-ISALT)
4526      F2=F0*(2-ITEMP)
4527      SN(L)=(S(L,N,M)*(1.-F1)+F1*S(L,NI,MI))
4528      TN(L)=(T(L,N,M)*(1.-F2)+F2*T(L,NI,MI))
4529      GOTO 200
4530 C      INFLOW CONDITIONS
4531  150   JFLOW=-2
4532      F0=FLOAT(ISPLIT)/FLOAT(MAX0(ISPLIT,NSTINF(L,NB)+NHRINF-NSTIT))
4533      F1=F0*(2-ISALT)
4534      F2=F0*(2-ITEMP)
4535      SN(L)=(S(L,N,M)*(1.-F1)+F1*SBND(L,NB))
4536      TN(L)=(T(L,N,M)*(1.-F2)+F2*TBND(L,NB))
4537  200   CONTINUE
4538  220   CONTINUE
4539 C      CHECK FOR POSITIVE VALUES
4540      IF(ICPOS.EQ.0)GOTO 240
4541      DO L=1,LBOT
4542      SN(L)=AMAXI(SN(L),0.0)
4543      TN(L)=AMAXI(TN(L),0.0)
4544      ENDO
4545  240   CONTINUE
4546      DO L=1,LBOT
4547      T(L,N,M)=TN(L)
4548      S(L,N,M)=SN(L)
4549      ENDO
4550  295   CONTINUE
4551  360   CONTINUE
4552  370   CONTINUE
4553      RETURN
4554      END
4555 C -----
4556 C -----
4557 C      SUBROUTINE BNDRY4
4558 C      NOVEMBER 1986 HESS & PYTLOWANY MEAD VAX
4559 C      PURPOSE - TO COMPUTE THE RIVERINE SALINITY AND TEMPERATURE
4560 C      BOUNDARY CONDITIONS
4561 C      INCLUDE 'COMM20.FOR'
4562 C      COMMON/BNDX4/ISALT,ITEMP,NB
4563 C      DIMENSION FSZ(LSIZE)
4564 C      DO L=1,LBOT
4565 C      FSZ(L)=2*(1.-FLOAT(L-1)/FLOAT(LBOT-1))
4566 C      ENDO
4567 C      IP=0
4568 C      IP=IPRN1
4569 C      RIVER FLOW BOUNDARIES
4570 C      IF(NUMRIV.LE.0)GOTO 300
4571      NB=0
4572      DO 250 NR=1,NUMRIV
4573      MPR1=NR
4574      MPR2=NR
4575      ML=MPR1-NR
4576      NF=NR1-NR
4577      NL=NR2-NR
4578      DO 220 M=MF,ML
4579      DO 220 N=NF,NL
4580      NB=NB+1
4581      IF(JTPR(NR).EQ.2)GOTO 130
4582 C      FLUME CONDITION
4583      DO 100 L=1,LBOT
4584      IF(ISALT.EQ.1)S(L,N,M)=SBND(L,NB+NBCEL0)
4585  100   IF(ITEMP.EQ.1)T(L,N,M)=TBND(L,NB+NBCEL0)
4586      GOTO 220
4587 C      WATER FALLS CONDITION
4588  130   SAL=SBND(1,NB+NBCEL0)
4589      TMP=TBND(1,NB+NBCEL0)
4590      KMAX=1+ML-MF+NL-NF
4591      ND=N
4592      MD=M
4593      VOL0=(D(ND,MD)+SE(ND,MD))*DQ*AREA(ND,MD)*DL**2
4594      VOL1=D1T1*RATE(NR)*DQ*FLOAT(RMAX)
4595      F3=VOL1/(VOL0+VOL1)
4596      IF(IP.EQ.1)WRITE(ISCR,140)NR,ND,MD
4597  140   FORMAT(1X,'RIVER: NR=',I2,' N,M=',2I4)
4598      IF(IP.EQ.1)WRITE(ISCR,*)"SAL,TMP=",SAL,TMP,' F3=',F3
4599      DO 170 L=1,LBOT
4600      F2=F3*FSZ(L)
4601      F1=1.-F2
4602      IF(ISALT.EQ.1)S(L,ND,MD)=(S(L,ND,MD)*F1+SAL*F2)
4603      IF(ITEMP.EQ.1)T(L,ND,MD)=(T(L,ND,MD)*F1+TMP*F2)
4604      IF(IP.EQ.1)WRITE(ISCR,160)L,S(L,ND,MD),T(L,ND,MD)
4605  160   FORMAT(3X,'L=',I3,' S=',F5.2,' T=',F5.2)
4606  170   CONTINUE
4607  220   CONTINUE
4608  250   CONTINUE
4609  300   RETURN
4610      END
4611 C -----
4612 C -----
4613 C      SUBROUTINE BSTATE
4614 C      FEBRUARY 1986 K. W. HESS CEOB SGI/IRIS
4615 C      PURPOSE - TO SET THE RIVERINE AND OCEANIC SALINITY AND
4616 C      TEMPERATURE BOUNDARY STATES BY INTERPOLATION
4617 C      VARIABLES -
4618 C      SBND(L,NB) = INTERPOLATED STATE OF BOUNDARY SALINITY AT
4619 C      LEVEL L AND BOUNDARY GRID NB
4620 C      TBND(L,NB) = INTERPOLATED STATE OF BOUNDARY TEMPERATURE
4621 C      NBC = TOTAL NUMBER OF OCEANIC AND RIVERINE
4622 C      BOUNDARY GRIDS (UP TO 100 ALLOWED)
4623 C      SET STANDARD CONDITIONS
4624      INCLUDE 'COMM20.FOR'
4625 C      COMMON/BNDX4/ISALT,ITEMP,NB
4626 C      DIMENSION SFINL(LSIZE),TFINL(LSIZE),TRFINL(NDRIV2)
4627 C      SET STANDARD CONDITIONS
4628      IF(IPRN1.EQ.1)WRITE(ISCR,100)UT,HRI,NSTI
4629  100   FORMAT(1X,'BSTATE : UT=',F10.4,' CUM HR=',F10.2,' NSTI=',I6)
4630 C      set to default values
4631      DO L=1,LBOT
4632      SFINL(L)=SAL0
4633      TFINL(L)=TMPO
4634      DO N=1,NM2SIZ
4635      SRND(L,N)=SAL0
4636      TBND(L,N)=TMPO
4637      ENDDO
4638      ENDDO
4639      DO N=1,NDRIV2
4640      TRFINL(N)=TMPO
4641      ENDDO
4642 C      initialize cell counts
4643      NBCEL=0
4644      NBCEL0=0
4645 C      check on ocean conditions
4646      IF(NUMOBC.LE.0)GOTO 300
4647 C      SET OCEANIC SALINITY CONDITIONS
4648      IF(KONCEN.EQ.2.OR.NSIGS.EQ.0.OR.IENDSO.EQ.1)GOTO 150
4649 C      read more salinity data
4650      CALL RR(YT,ISCR,LUSAL,IENDSO,DSAL,YSAL,NSIGS,LSIZE,SALOCN,SFINL)
4651 C      OCEAN TEMP
4652  150   IF(KONCEN.EQ.1.OR.NSIGT.EQ.0.OR.IENDTO.EQ.1)GOTO 165
4653 C      read more ocean temperature data
4654      CALL RR(YT,ISCR,LUCR,IENDTO,DOTF,YOTP,NSIGTO,LSIZE,TMPOCN,TFINL)
4655  165   IF(IPRN1.EQ.0)GOTO 200
4656      DO L=1,LBOT
4657      WRITE(ISCR,170)L,SFINL(L),TFINL(L)
4658  170   FORMAT(3X,'L=',I2,' SFINL=',F7.3,' TFINL=',F7.3)
4659      ENDDO
4660 C      LOOP THRU OCEANIC BOUNDARIES
4661  200   DO 250 IB=1,NUMOBC
4662 C      LOOP THRU EACH GRID
4663  250   DO 250 N=NB1(IB),NB2(IB)
4664  250   DO 250 M=MB1(IB),MB2(IB)
4665      NBCEL=NBCEL+1
4666      NBCEL0=NBCEL0+1
4667 C      SET VALUES OVER DEPTH
4668  240   DO 240 L=1,LBOT
4669      SBND(L,NBCEL)=SFINL(L)
4670  240   TBND(L,NBCEL)=TFINL(L)
4671  250   CONTINUE
4672 C      RIVER BOUNDARIES
4673  300   IF(NUMRIV.LE.0)GOTO 400
4674  300   read more river temperature data
4675 C      IF(KONCEN.EQ.1.OR.NSIGRT.EQ.0.OR.IENDRT.EQ.1)GOTO 310
4676      CALL RR(YT,ISCR,LURV,IENDRT,DRVT,YRVT,NSIGRT,NDRIV2,TRIV,TRFINL)
4677 C      LOOP THRU THE RIVERS
4678  310   DO 340 NR=NR1,NR2
4679  310   DO 340 NR=NR1(NR),NR2(NR)
4680 C      LOOP THRU THE GRIDS
4681  340   DO 340 M=MR1(NR),MR2(NR)
4682  340   DO 340 N=NR1(NR),NR2(NR)
4683      NBCEL=NBCEL+1
4684 C      LOOP OVER DEPTH
4685  320   DO 320 L=1,LBOT
4686      SBND(L,NBCEL)=0.0
4687  320   TBND(L,NBCEL)=TRFINL(NR)
4688  330   IF(IPRN1.EQ.1)WRITE(ISCR,330)NR,TRFINL(NR)
4689  330   FORMAT(3X,'NR=',I2,' TRFINL=',F7.3)
4690  340   CONTINUE
4691  400   CONTINUE
4692      RETURN
4693      END
4694      END
4695 C -----
4696 C -----
4697 C      SUBROUTINE GFLUX
4698 C      JUNE 1996 K. W. HESS
4699 C      PURPOSE - TO COMPUTE NEW DISTRIBUTION OF CONCENTRATE,
4700 C      WHICH INCLUDES VARIABLE WIDTH AND VARIABLE HORIZONTAL
4701 C      VISCOSITY. NEW FORMULATION OF UPPER BOUNDARY
4702 C      CONDITION.
4703 C      VARIABLES
4704 C      SN() = NEW (UPDATED) SALINITY
4705 C      SC() = SALINITY AT ROW AT START OF UPDATE
4706 C      SM() = SALINITY AT PREVIOUS ROW AT START OF UPDATE
4707 C      FA(),FB() = RECURSION ARRAYS FOR SALINITY, TEMPERATURE
4708 C      FTOP = FLUX OF C AT THE AIR-WATER INTERFACE
4709 C
4710 C      711 C      L-1 + U(L-1), W, S
4712 C      712 C      |-----| DV(L-1) -----| ULM, (DM,GM), FFZA(L-1)
4713 C      713 C      |-----| DV(L) -----| FFCC(L)
4714 C      714 C      |-----| L + U(L), W, S
4715 C      715 C      |-----| DV(L) -----| ULP, DP, GP, FFZA(L), FFZD(L)
4716 C      716 C      |-----| DV(L) -----| C1=DT1/(4.*DQ)
4717 C      717 C      |-----| DV(L) -----| C2=DT1/(DQ*2)
4718 C      718 C      |-----| DV(L) -----| C3=DHAI*DT1/(2.*DL**2)
4719 C      719 C      |-----| DV(L) -----| C4=DT1/(2.*DL)
4720 C      720 C      |-----| DV(L) -----| C5=DT1/(16.*DL)
4721 C      INCLUDE 'COMM20.FOR'
4722 C      COMMON/BNDX4/ISALT,ITEMP,NB
4723 C      DIMENSION SN(LSIZE),SC(LSIZE,NSIZE),SM(LSIZE,NSIZE),
4724  2   TMP(LSIZE),TC(LSIZE,NSIZE),TM(LSIZE,NSIZE),TRAD(LSIZE),
4725  2   FFCC(LSIZE),FFXM(LSIZE),FFXP(LSIZE),FFYM(LSIZE),FFYP(LSIZE),
4726  3   FFZA(LSIZE),FFZD(LSIZE),DEN(LSIZE),PROD(LSIZE)
4727      DATA IB/0/
4728      C1=DT1/(4.*DQ)
4729      C2=DT1/(DQ*2)
4730      C3=DHAI*DT1/(2.*DL**2)
4731      C4=DT1/(2.*DL)
4732      C5=DT1/(16.*DL)
4733      C5=DT1/(16.*DL)
4734      C5=DT1/(16.*DL)
4735 C      INITIALIZE SURFACE HEAT FLUX
4736      IF(KONCEN.EQ.2)CALL HEAT1
4737 C      SOLVE GENERALIZED TRANSPORT EQUATION
4738 C      LOOP DOWN THE LINES
4739 C
4740 C      740 C      DO 400 M=1,MMAX
4741      DO 400 M=1,MMAX
4742 C      STORE VALUES
4743  400   DO 130 N=1,NMAX
4744  130   DO 130 L=1,LBOT
4745  130   IF(M.GT.1)GO TO 110
4746  110   SM(N-1)=SC(L,N,M)
4747  110   TM(L,N)=TC(L,N,M)
4748  110   GOTO 120
4749  110   SM(L,N)=SC(L,N)
4750  110   TM(L,N)=TC(L,N)
4751  120   SC(L,N)=S(L,N,M)
4752  120   TC(L,N)=T(L,N,M)
4753  130   CONTINUE
4754      RUN ACROSS ROW
4755 C      4755 C      NA=NAB(M)/1000
4756 C      4756 C      NB=MOD(NAB(M),1000)
4757 C      4757 C      IF(NA.GT.NB)GOTO 400
4758 C      4758 C      MM=MAX0(M-1,1)
4759 C      4759 C      MP=MIN0(MMAX,M+1)
4760      MP=MIN0(MMAX,M+1)

```

```

4761 C      LOOP ACROSS ROWS
4762 DO 390 N=NR,NB
4763 IF(IFIELD(N,M).LT.10.OR.IFIELD(N,M)/10.EQ.KOCNHC)GOTO 390
4764 NM=MAX0(N-1,1)
4765 NP=MINO(NMAX,N+1)
4766 C      COMPUTE REPEATEDLY-USED TERMS
4767 H=D(N,M)*SE(N,M)+E
4768 ANN=AREA(N,M)
4769 F1=C1*ANN
4770 F2=C2*ANN/H
4771 HH=ANN*H
4772 SS=ANN*(SE(N,M)-SOLD(N,M))
4773 C      HORIZONTAL X-DIRECTION FLUX TERMS
4774 UH01=BX(N,M-1)*C4*UHOLD(N,M-1)
4775 H1=BX(N,M-1)*C5*(H+D(N,M-1)+SE(N,M-1))
4776 DH1=BX(N,M-1)*C3*FLOAT(MFLUX(N,M))
4777 UH02=BX(N,M)*C4*UHOLD(N,M)
4778 H2=BX(N,M)*C5*(H+D(N,M-1)+SE(N,M-1))
4779 DH2=BX(N,M)*C3*FLOAT(MFLUX(N,M))
4780 C      HORIZONTAL Y-DIRECTION FLUX TERMS
4781 VHO1=(N-1,M)*C4*VHOLD(N-1,M)
4782 H3=BY(N,M-1)*C5*(H+D(N,M-1)+SE(N,M-1))
4783 DH3=BY(N,M-1)*C3*FLOAT(NFLUX(N,M))
4784 VHO2=BY(N,M)*C4*VHOLD(N,M)
4785 H4=BY(N,M)*C5*(H+D(N+1,M)+SE(N+1,M))
4786 DH4=BY(N,M)*C3*FLOAT(NFLUX(N,M))
4787 C      INITIAL VELOCITIES
4788 C      L=LBOT
4789 ULP1=U(L,N,M-1)+U(L-1,N,M-1)
4790 ULP2=U(L,N,M)+U(L-1,N,M)
4791 VLP1=V(L,N-1,M)+V(L-1,N-1,M)
4792 VLP2=V(L,N,M)+V(L-1,N,M)
4793 C      LOOP THRU THE LAYERS, STARTING AT BOTTOM
4794 DO 150 L=LBOT,1,-1
4795 LM=MAX0(1,L-1)
4796
4797 C      VERTICAL ADVECTIVE AND DIFFUSIVE FLUX TERMS: units are m/s
4798 FFZA(L)=F1*(W(L,N,M)+W(L+1,N,M)) ! Gp
4799 FFZD(L)=F2*DV(L,N,M) ! Dp
4800 C      GET NEW VELOCITIES
4801 LEM=L-1
4802 IF(L.EQ.1)LEM=L-1
4803 ULM1=U(L,N,M-1)+UH1*(AH3(L,N,M)+AH3(L,N,M-1))
4804 ULM2=U(L,N,M)+U(LM,N,M)
4805 VLM1=V(L,N,M)+V(LM,N-1,M)
4806 VLM2=V(L,N,M)+V(LM,N,M)
4807 C      GET HORIZONTAL GENERALIZED FLUX COEFFICIENTS : units are meters
4808 FFXM(L)=(H1*(ULM1+ULP1))+UH01*+DH1*(AH3(L,N,M)+AH3(L,N,M-1))
4809 FCC(L)=(H1*(ULM1+ULP1))+UH01*-DH1*(AH3(L,N,M)+AH3(L,N,M-1))
4810 1 (H2*(ULM2+ULP2))+UH02*-DH2*(AH3(L,N,M)+AH3(L,N,M-1))
4811 2 +H3*(VLM1+VLP1))+VHO1*-DH3*(AH3(L,N,M)+AH3(L,N-1,M))
4812 3 -H4*(VLM2+VLP2))+VHO2*-DH4*(AH3(L,N,M)+AH3(L,N-1,M))
4813 4 +HH
4814 FFPX(L)=1./*(H2*(ULM2+ULP2))+UH02)+DH2*(AH3(L,N,M)+AH3(L,N,M+1))
4815 FFYM(L)=(H3*(VLM1+VLP1))+VHO1*+DH3*(AH3(L,N,M)+AH3(L,N-1,M))
4816 FFPY(L)=1./*(H4*(VLM2+VLP2))+VHO2)+DH4*(AH3(L,N,M)+AH3(L,N+1,M))
4817 C      SAVE UPPER (LM) VELOCITIES IN LOWER (LP) VELOCITIES
4818 ULP1=ULM1
4819 ULP2=ULM2
4820 VLP1=VLM1
4821 VLP2=VLM2
4822 150 CONTINUE
4823 FFZA(LBOT)=0.0
4824 FFZD(LBOT)=0.0
4825 C      LOOP UP COLUMN, COMPUTE REPEATED QUANTITIES
4826 DEN(LBOT)=1./*(HH+SS+2.* (FFZA(LAYRS)+FFZD(LAYRS)))
4827 FB(LBOT)=2.* (FFZD(LAYRS)-FFZA(LAYRS))*DEN(LBOT)
4828 DO 160 L=LAYRS,2,-1
4829 LM=L-1
4830
4831 DEN(L)=1./*(HH+SS+2.* (FFZD(L)-FFZA(L))+FFZD(L)-FB(L+1)*
4832 1 (FFZA(L)+FFZD(L)))
4833 FB(L)=(FFZD(L)-FFZA(L))*DEN(L)
4834 160 CONTINUE
4835 DEN(1)=1./*(HH+SS+2.* (FFZD(1)-FFZA(1))-FB(2)*(FFZA(1)+FFZD(1)))
4836 C      SALINITY CALCULATIONS
4837 GOTO(200,250),ISALT
4838 200 CONTINUE
4839 C      BOTTOM CONDITIONS
4840 C      BOTTOM CONDITIONS
4841 L=LBOT
4842 CFLUX=FFXM(L)*SM(L,N)
4843 1 +FFYM(L)*SC(L,N)+FCC(L)*SC(L,N)+FFYP(L)*SC(L,NP)
4844 2 +FFXP(L)*S(L,N,MP)
4845 FA(L)=CFLUX*DEN(L)
4846 C      CORE SALINITY
4847 DO 230 L=LAYRS,2,-1
4848 CFLUX=FFXM(L)*SM(L,N)
4849 1 +FFYM(L)*SC(L,N)+FCC(L)*SC(L,N)+FFYP(L)*SC(L,NP)
4850 2 +FFXP(L)*S(L,N,MP)
4851 FA(L)=(CFLUX+FA(L+1))*(FFZA(L)+FFZD(L)))*DEN(L)
4852 230 CONTINUE
4853 C      TOP CONDITIONS
4854 L=1
4855 CFLUX=FFXM(L)*SM(L,N)
4856 1 +FFYM(L)*SC(L,N)+FCC(L)*SC(L,N)+FFYP(L)*SC(L,NP)
4857 2 +FFXP(L)*S(L,N,MP)
4858 S(1,N,M)=(CFLUX+2.*FA(2)*(FFZA(1)+FFZD(1)))*DEN(1)
4859 C      INVERT HERE
4860 DO L=2,LBOT
4861 S(L,N,M)=FA(L)+FB(L)*S(L-1,N,M)
4862 ENDDO
4863 C      TEMPERATURE
4864 C      TOP AND BOTTOM UPWARD HEAT FLUX
4865 250 GOTO(260,390),ITEMP
4866 C      BOTTOM CONDITIONS
4867 260 CALL HEATZ(N,M,TRAD,FTSURF)
4868 C      BOTTOM CONDITIONS
4869 FTBOT=0. ! heat lost into bottom sediment
4870 L=LBOT
4871 TFLUX=FFXM(L)*TM(L,N)
4872 1 +FFYM(L)*TC(L,NM)+FCC(L)*TC(L,N)+FFYP(L)*TC(L,NP)
4873 2 +FFXP(L)*T(L,N,MP)
4874 FA(LBOT)=(TFLUX+HH*(TRAD(L)+FTBOT))*DEN(L)
4875 C      CORE TEMPERATURE
4876 DO 300 L=LAYRS,2,-1
4877 TFLUX=FFXM(L)*TM(L,N)
4878 1 +FFYM(L)*TC(L,NM)+FCC(L)*TC(L,N)+FFYP(L)*TC(L,NP)
4879 2 +FFXP(L)*T(L,N,MP)
4880 FA(L)=(TFLUX+HH*TRAD(L)+FA(L+1)*(FFZA(L)+FFZD(L)))*DEN(L)
4881 300 CONTINUE
4882 C      TOP TEMPERATURES
4883 L=1
4884 TFLUX=FFXM(L)*TM(L,N)
4885 1 +FFYM(L)*TC(L,NM)+FCC(L)*TC(L,N)+FFYP(L)*TC(L,NP)
4886 2 +FFXP(L)*T(L,N,MP)

4887 T(1,N,M)=(TFLUX+HH*(TRAD(1)+FTSURF))+2.*FA(2)*(FFZA(L)
4888 1 +FFZD(L)))/(HH+SS+2.* (FFZD(L)-FFZA(L))-FB(2)*(FFZA(L)+FFZD(L)))
4889 C      INVERT HERE
4890 DO L=2,LBOT
4891 T(L,N,M)=FA(L)+FB(L)*T(L-1,N,M)
4892 ENDDO
4893 390 CONTINUE
4894 400 CONTINUE
4895 C      make positive
4896 IF(ICPOS.EQ.0)RETURN
4897 DO 600 M=1,MMAX
4898 DO 600 N=1,NMAX
4899 S(L,N,M)=AMAX1(S(L,N,M),0.0)
4900 600 T(L,N,M)=AMAX1(T(L,N,M),0.0)
4902 RETURN
4903 END
4904 C
4905 C-----SUBROUTINE HEAT1
4906 C      APRIL 1985 K. W. HESS MEAD VAX 11/750
4907 C      PURPOSE - TO COMPUTE OCEAN SURFACE HEAT FLUX CONSTANTS FOR
4908 C      THIS TIMESTEP FOR ALL CELLS
4909 C      INPUTS -
4910 C      IHEAT = HEAT FLUX INDEX:
4911 C      0=NO HEAT,
4912 C      1=HEAT FLUX BUT WITH DAY OF YEAR CONSTANT (=NDAY
4913 C      2=NORMAL HEATING
4914 C      CLOUD = FRACTION OF SKY COVERED BY CLOUDS (0.0 - 1.0)
4915 C      TAIRK = AIR TEMPERATURE (K) AT 10 M
4916 C      W10 = WIND SPEED AT 10 M (M/S)
4917 C      PA = ATMOSPHERIC PRESSURE (PASCALS)
4918 C      RELHUM = RELATIVE HUMIDITY AT 10 M (0.0 - 1.0)
4919 C      TW = WATER TEMPERATURE AT NEAR-SURFACE (K)
4920 C      VARIABLES -
4921 C      COSZ = COSINE OF Z
4922 C      CORGAW = AIR-WATER INTERFACIAL DRAG COEFFICIENT
4923 C      DEC = SUN'S DECLINATION (ANGLE ABOVE ECLIPTIC)
4924 C      EA, ES = SATURATION VAPOR PRESSURE (PASCALS)
4925 C      HSOLAR = HOUR IN DAY ON A 24-HOUR CLOCK
4926 C      NETDAY = DAY OF THE YEAR, COUNTED FROM JANUARY 1.
4927 C      TO = AIR TEMPERATURE AT SEA SURFACE (K)
4928 C      CONSTANTS -
4929 C      ALB = NET ALBEDO
4930 C      ALV = LATENT HEAT OF VAPORIZATION (2.6 X 10 6 J/KG)
4931 C      RHQA = AIR DENSITY (KG/M3)
4932 C      RHOW = SEA WATER DENSITY (KG/M3)
4933 C      CPAIR = SPECIFIC HEAT CAPACITY OF DRY AIR (J/KG/K)
4934 C      CWATER = SPECIFIC HEAT CAPACITY OF WATER (J/KG/K)
4935 C      SB = STEFAN-BOLTZMAN CONSTANT
4936 C      INCLUDE 'COMM20.FOR'
4937 C      DIMENSION VAL(4)
4938 C      DATA NDAY/180/,IP0/
4939 C      CLOUDNESS FUNCTION
4940 C      FCC(X)=1.-X
4941 C      T2KEL(TEMP)=TEMP+273.*(.5-SIGN(.5,TEMP-200.))
4942 C      GET DATA
4943 C      IF(NSIGM,EQ.0)THEN
4944 C      TAIRC=20.
4945 C      RELHUM=.7
4946 C      CLOUD=.5
4947 C      PA=1014.0*100.
4948 C      ELSE
4949 C      CALL RE(YT,ISCR,LUMET,IENDMT,DMET,YMET,NSIGM,4,VMET,VAL)
4950 C      TAIRC=VAL(1)
4951 C      RELHUM=VAL(2)
4952 C      CLOUD=VAL(3)
4953 C      PA=VAL(4)*100.
4954 C      ENDIF
4955 C      IF(IHEAT.EQ.1)NDAY=NDAY0
4956 C      IF(IHEAT.EQ.2)NDAY=UT
4957 C      IF(IPRNT1,EQ.1)WRITE(ISCR,100)TAIRC,RELHUM,CLOUD,PA,NDAY
4958 C      100 FORMAT(1X,'HEAT1: TAIRC=',F5.2,' RELHUM=',F5.3,' CLOUD=',F4.2,
4959 C      1 ' PA=',F10.2,' NDAY=',I3)
4960 C      NEAR-SURFACE AIR TEMPERATURE
4961 C      TAIRC=TAIRC+273.
4962 C      T10=TAIRC
4963 C      SOLAR ANGLES
4964 C      HSOLAR=24.*UT-FLOAT(IFIX(UT))-5.*DTI/3600.
4965 C      DEC=23.44*COS(FLOAT(172-NDAY)*PI/182.5)
4966 C      COSZ=SIN(BNLAT*RAD)*SIN(DEC*RAD)+COS(BNLAT*RAD)*COS(DEC*RAD)*
4967 C      1 COS((12.-HSOLAR)*PI/12.)
4968 C      CONSTANTS
4969 C      CSOL1=FCC(CLOUD)*(1.-ALB)*SOLAR*(COSZ**2)
4970 C      CSOL2=0.101*.085*COSZ
4971 C      CSOL3=(COSZ+2.7)/(10**5)
4972 C      ES=1.0-1.0**4*(7.5*(T10-273.16)/(T10-35.86))
4973 C      EA=RELHUM**ES
4974 C      CEVP2=EA/(PA-(1.-EPSILON)*EA)
4975 C      CTOT1=1./((HOW*CWATER)
4976 C      CH1=SB*CTOT1
4977 C      ATMOSPHERIC LONG WAVE INWARD
4978 C      A=MIN0(1,IHEAT)
4979 C      QA=A*SB*TAIRC**4*(1.-0.261*EXP(-0.000777*(273.-TAIRK)**2))
4980 C      IF(IPRNT1,EQ.1)WRITE(ISCR,200)CSOL1,CSOL2,CSOL3
4981 C      200 FORMAT(1X,'COS2,CSOL1,2,3=',4E10.2)
4982 C      RETURN
4983 C      END
4984 C-----SUBROUTINE HEATZ(N,M,TRAD,FTSURF)
4985 C      APRIL 1985 K. W. HESS MEAD VAX 11/750
4986 C      PURPOSE - TO COMPUTE OCEAN SURFACE HEAT FLUXES INTO A
4987 C      PARTICULAR CELL
4988 C      VARIABLES -
4989 C      CSENS = COEFFICIENT FOR SENSIBLE HEAT GAIN (SEE HEAT1)
4990 C      EA, ES = SATURATION VAPOR PRESSURE (PASCALS)
4991 C      EPSLON = RATIO OF MOLECULAR WEIGHTS OF VAPOR AND DRY AIR
4992 C      QI,QA,QB,QE,OS = heat inputs (Watts/m**2/K**4)
4993 C      RELHUM = RELATIVE HUMIDITY (0.0 < RELHUM < 1.0)
4994 C      SB = STEFAN-BOLTZMAN CONSTANT (Watt/meter**2/K**4)
4995 C      TAIRK = AIR TEMPERATURE (K)
4996 C      TRAD = AIR TEMPERATURE AT SEA SURFACE (K)
4997 C      T = AIR TEMPERATURE AT NEAR-SURFACE (K)
4998 C      TRAD = CHANGE IN TEMPERATURE OVER ONE TIMESTEP (DEG C)
4999 C      TW = WATER TEMPERATURE AT NEAR-SURFACE (K)
5000 C      INCLUDE 'COMM20.FOR'
5001 C      DIMENSION TRAD(ILSIZE)
5002 C      DATA IP0/
5003 C      DATA IP10/
5004 C      DATA IP11/
5005 C      DATA IP12/
5006 C      DATA IP13/
5007 C      DATA IP14/
5008 C      DATA IP15/
5009 C      DATA IP16/
5010 C      DATA IP17/
5011 C      DATA IP18/
5012 C      IP1=0

```

```

5013 IF(NPRMN,EQ.0) GOTO 100
5014 DO J=1,NPRMN
5015 IF(IP1.EQ.1.AND.IPRNT1.EQ.1.AND.N+1000*M.EQ.IPRMN(J)) IP1=1
5016 ENDDO
5017 100 CONTINUE
5018 C      SET CONSTANT
5019 A=MNO(1,HEAT)
5020 C      WATER SURFACE TEMP (C)
5021 TWS=T2KEI(.5*(T1,N,M)+T(2,N,M))
5022 C      SET AIR TEMP JUST ABOVE SURFACE (K) EQUAL TO WATER TEMP
5023 FBETA=.9
5024 TO=FBETA*TWS+(1-FBETA)*TAIRK
5025 C      BULK TRANSFER COEFFICIENT
5026 W10=SQRT(WX(N,M)**2+WY(N,M)**2)
5027 CDRGAW=0.0025*SIGN(.0015,TWS-TAIRK) ! HIGHER IF UNSTABLE (TW > TA
5028 CSENS=RHOA*CDRGAW*CPAIR*AMAX1(W10,0.2)
5029 CEVPI=ALV*RHOA*CDRGAW*EPSILON*AMAX1(W10,0.2)
5030 C      print
5031 IF(IP1.EQ.1) WRITE(ISCR,105) UT,NM
5032 105 FORMAT(1X,'HEATZ: UT=',F10.4,' NM, M=',2I4)
5033 IF(IP1.EQ.1) WRITE(ISCR,110)TAIRK,TWS,T0,W10
5034 110 FORMAT(1X,'TAIRK, TWS, T0 (K)=',3F8.2,' W10=',F6.2)
5035 C      SATURATION VAPOR PRESSURE AT AIR-WATER INTERFACE
5036 ES=611.0**T/(7.5*(T0-273.16)/(T0-35.86))
5037 EA=RELHUM**ES
5038 C      SOLAR SHORT-WAVE INWARD (Watt/m**2)
5039 Q1=0.0
5040 IF(COSZ.GT.0.0) Q1=AMAX1(0.0,CSOL1/(CSOL2+CSOL3*EA))
5041 C      WATER BLACK BODY INWARD
5042 QB=EA*(.97)*SB*(TWS)**4
5043 C      SENSIBLE HEAT GAIN
5044 QS=AMAX1(TAIRK-TWS)
5045 C      EVAPORATIVE HEAT GAIN
5046 QE=AMAX1(CEVPI*2-EA/(PA-(1.-EPSILON)*EA))
5047 QSUM=QA+QS+QE+QB
5048 IF(IP1.EQ.1) WRITE(ISCR,130)QB,QS,QA,QE,QSUM
5049 130 FORMAT(1X,'QB,QS,QA,QE,QSUM=',5E10.3)
5050 C      TOTALS: FQ, FS HAVE UNITS (DEG C)*M/S   CTOT=1./((RHO*CWATER)
5051 FQ=CTOT*QSUM
5052 C      SURFACE LAYER HEATING UNITS=DEG C
5053 FTSURF=F0*DTI/((D(N,M)+SE(N,M))*HALFDQ)
5054 IF(IP1.EQ.1) WRITE(ISCR,*)'FTSURF=',FTSURF
5055 C      INTERNAL SOLAR HEATING: TRAD units= deg C
5056 FS=CTOT*QS ! deg C x m/s
5057 H=D(N,M)*TSE(N,M)
5058 IF(IP1.EQ.1) WRITE(ISCR,*)'FS=',FS, ' H=',H
5059 FD10=2.0/D10PCT
5060 DO 220 L=1,LBOT
5061 TRAD(L)=0.0
5062 ZTOP=H*AMIN1(0.0,DQ*FLOAT(1-L)+HALFDQ)
5063 IF(ZTOP.LT.-2.*D10PCT) GOTO 220
5064 ZBOT=H*AMAX1(-1.,DQ*FLOAT(1-L)-HALFDQ)
5065 TRAD(L)=FS*DTI*(EXP(FD10*ZTOP)-EXP(FD10*ZBOT))/(ZTOP-ZBOT)
5066 IF(IP1.EQ.1) WRITE(ISCR,150)L,ZTOP,TRAD(L)
5067 150 FORMAT(1X,'L=',I2,' ZTOP=',F6.2,' TRAD=',E10.4)
5068 220 CONTINUE
5069 RETURN
5070 END
5071 C-----SUBROUTINE SETSTP
5072 C-----SUBROUTINE SETBND
5073 C-----SUBROUTINE SETOCEAN
5074 C-----SUBROUTINE SETRIVER
5075 JUNE 1985 K. W. HESS MEAD VAX 11/750
5076 C PURPOSE - SET UP THE INITIAL FIELDS OF SALINITY AND
5077 C TEMPERATURE BY INTERPOLATING FROM THE BOUNDARY
5078 C CONDITIONS. SET INITIAL HORIZONTAL PRESSURE GRADIENTS
5079 C VARIABLES - ICS = INDEX FOR READING INITIAL CONDITIONS
5080 C           (0=NO, 1=YES)
5081 C           NBCELL = TOTAL NUMBER OF BOUNDARY GRIDS (RIV + OCEAN)
5082 C INCLUDE 'CM200.FOR'
5083 C DIMENSION NO(NM2SIZ),MO(NM2SIZ),R(NM2SIZ),SBNDP(LSIZE,NM2SIZ)
5084 C SET BOUNDARY STATE
5085 CALL BSTATE
5086 IF(ICS.EQ.1) GOTO 310
5087 C      SET DEFAULT VALUES
5088 DO 100 L=1,LBOT
5089 100 N=1,NM2SIZ
5090 SBNDP(L,N)=SBND(L,N)
5091 DO 120 N=N1,I,NB2(I)
5092 SBNDP(L,N)=SBND(L,N)
5093 IF(N.GT.NBECO.AND.KONCEN.EQ.2) SBNDP(L,N)=SALO
5094 IF(N.GT.NECEO.AND.MOD(KONCEN+1,2).EQ.0) SBNDP(L,N)=5.
5095 100 continue
5096 C      SET OCEANIC BOUNDARY VALUES
5097 J=0
5098 IF(NUMOBC.LE.0) GOTO 200
5099 DO 120 I=1,NUMOBC
5100 DO 120 N=N1(I),NB2(I)
5101 DO 120 M=MB1(I),MB2(I)
5102 J=J+1
5103 MO(J)=M
5104 NO(J)=N
5105 DO 120 L=1,LBOT
5106 S(L,N,M)=SBNDP(L,J)
5107 T(L,N,M)=TBND(L,J)
5108 120 CONTINUE
5109 C      SET RIVER BOUNDARY VALUES
5110 130 IF(NUMRIV.LE.0) GOTO 200
5111 DO 140 I=1,NUMRIV
5112 DO 140 M=MR1(I),MR2(I)
5113 DO 140 N=NR1(I),NR2(I)
5114 J=J+1
5115 MO(J)=M
5116 NO(J)=N
5117 DO 140 L=1,LBOT
5118 S(L,N,M)=SBNDP(L,J)
5119 T(L,N,M)=TBND(L,J)
5120 140 CONTINUE
5121 C      LOOP THRU THE INTERIOR COMPUTATIONAL GRIDS
5122 200 DO 300 M=1,MMAX
5123 DO 300 N=1,NMAX
5124 IF(IFIELD(N,M).LT.10.OR.IFIELD(N,M).GE.10*(KOCNBC)) GOTO 300
5125 IF(D(N,M).LE.0.0) WRITE(ISCR,240)N,M
5126 240 FORMAT(1X,'** SETUP: NO DEPTH AT N=',I3,', M=',I3)
5127 C      FIND RADAI TO BOUNDARY DATUM, AND SUM OF RADAI
5128 RSUM=0.0
5129 DO 250 I=1,NBCEL
5130 R(I)=1.0/(2.25*(FLOAT((N-NO(I))*2)+FLOAT((M-MO(I))*2)))
5131 250 RSUM=RSUM+R(I)
5132 255 RINV=1./RSUM
5133 DO 270 L=1,LBOT
5134 S(L,N,M)=0.0
5135 T(L,N,M)=0.0
5136 DO 260 I=1,NBCEL
5137 S(L,N,M)=S(L,N,M)+(R(I)*RINV)*SBNDP(L,I)
5138 T(L,N,M)=T(L,N,M)+(R(I)*RINV)*TBND(L,I)
5139 260 CONTINUE
5140 270 CONTINUE
5141 300 CONTINUE
5142 310 CONTINUE
5143 RETURN
5144 END
5145 C-----COMMS FOR - COMMON BLOCKS FOR PROGRAM MECCA
5146 C      PARAMETER (NSIZE = 34, MSIZE = 55, LSIZE = 10, NMSIZE=NSIZE*MSIZE,
5147 1 LNMSIZ=LSIZE*NSIZE*MSIZE, NPMISIZ=NSIZE+MSIZE+2, NM2SIZ=2*(NSIZE+
5148 2 MSIZE), NDID2=5, NDIV2=10, NDCN2=LSIZE, NDMET2=4)
5149 C      MODEL CONSTANTS
5150 C-----COMMON/CONS1/A,OMEGA,PI,RAD,E,VONKAR,CRICH(8),RIMIN,RIMAX,CRO,
5151 1 IEXTRN,COR,NCOR,MCOR,DFDN,DFDM,BSNLAT,BSNANG,NONLIN,DL,CDRGWB,
5152 2 FCORO,VERS,FEDGE(NSIZE,MSIZE),IHSE,CWB1,CWB2,RAMP,IGRADP,
5153 3 RAMPT,RAFW,IBETAA,IBETAP,IBETAH,RHOBAR,INTER,SALO,TMPO,CSO,CS1,
5154 4 CST,CT1,CT2,ISTOP,ICS,KTEST,NDAYMO(12),IVISCV,AH0,AH0,CAH,RHOA,
5155 5 RHOW,DHAB,DTE,DTI,ISPRINT,NSTIT,NSTE,NSTET,NSTIM,DTAUI,DTAU2
5156 6 RHOW,DHAB,DTE,DTI,ISPRINT,NSTIT,NSTE,NSTET,NSTIM,DTAUI,DTAU2
5157 C      OUTPUT FORMAT VARIABLES
5158 PARAMETER (NDPRF=20, NDLC2=20, NDGPB=20)
5159 CHARACTER*10 CTITLE,PTITLE
5160 REAL 8
5161 COMMON/TIME/NSTO,UT,UTO,UT1,HR,HRO,HR1,IHR,YT,IYEAR,MONTH,
5162 1 IDAY,IHOUR,IMIN,YEAR,YEAR1,CUNDAY,NMAX,MMAX,NCELL,DMAX,
5163 2 NEGS,ICHECK,ICOR,IBOTV,ITOPV,IVISCV,NVISCV,IBRNT1,JPRINT(15),
5164 3 KRNTP1,KRNTP2,HRRM,HROUT,HROUTN,IPRMN,IPRN1(NDRN)
5165 4 CTITLE(8),ISLICE,JSLICE(NDSL1),NSLICE(NDSL2,NDSL1),MSLICE(
5166 5 NDSL2,NDSL1),ISTART,NGPMAX,IGPH,NSTGPB,IGPHB,HSAVE,
5167 6 LGPH(NDGPB),MGPH(NDGPB),NGPH(NDGPB),ITYP(NDGPB),PTITLE(25)
5168 C      MODEL GRID AND GEOGRAPHY VARIABLES
5169 COMMON/CON2/BSNLN,CON2M,HSML,NUMBIV,NCOL1,IFIELD(NSIZE,MSIZE),
5170 1 ICOL(5,NM2SIZ),IROW(5,NM2SIZ),NCOL,NROW,NAB(NM2SIZ),DMIN,KOCNCC,
5171 2 KRIVBC,IBAR,GRX(LSIZE),GRY(LSIZE),IHEAT,NISIGW,DWND(2),YWND(2),
5172 3 BX(NSIZE,MSIZE),BY(NSIZE,MSIZE),THETA1(NSIZE,MSIZE),
5173 4 THETA2(NSIZE,MSIZE),THETA3(NSIZE,MSIZE),NUMXY,20
5174 C      TIDE, MET, AND RIVER VARIABLES
5175 COMMON/TIDES1/NSIGT,IENDTD,YTID(2),DTID(2),TDLEV(2,NDTID2),
5176 1 IENDWN,WX(NSIZE,MSIZE),WY(NSIZE,MSIZE),DPADX,DPADY,
5177 2 FX(2,NSIZE,MSIZE),FY(2,NSIZE,MSIZE),DENRAT,CDR1,CDR2,TAIRK,TAIRC,
5178 3 TSX(NSIZE,MSIZE),TSY(NSIZE,MSIZE),IHEAT,NISIGW,DWND(2),YWND(2),
5179 4 FA,CLOUD,RELHUM,NSIGM,DMET(2),YMET(2),VMET(2,NDMET2),INDMT,
5180 5 NSIGR,YR1(2),DRV1(2),QRIV(2,NDRIV2),RATE(NDRIV2),ISETR(NDRIV2),
5181 6 IENDR,TRIV(NDRIV2),NDRIV2,TRIV(NDRIV2),NDRIV2,INDR,
5182 7 ITFR(NDRIV2),JTFR(NDRIV2),NSIGT,YRVT(2),DRV1(2),TRIV(2),NDRIV2
5183 C      OCEAN BOUNDARY CONDITIONS
5184 COMMON/FLAG1/NBCL1,NB21(NDCN2),NB2(NDCN2),NB1(NDCN2),NB2(NDCN2),
5185 1 NDCN2),ITPO(NDCN2),JTFP(NDCN2),ISET1(NDCN2),ISET2(NDCN2),
5186 2 NSIGS,YSAL(2),DSAL(2),SALON(2,NDCN2),IENDSO,NSIGTO,YOTF(2),
5187 3 DOTP(2),TMPOCN(2,NDCN2),IENDTO,NBCEL,NBCOLO,SBND(LSIZE,NM2SIZ),
5188 4 TBND(LSIZE,NM2SIZ),NSTINF(LSIZE,NM2SIZ),ITFO2(NDCN2),
5189 C      INTERNAL MODE (3-D) VARIABLES
5190 COMMON/TRIDI/LAYRS,LBOT,DQ,HALFDQ,TWODO,AH3(LSIZE,NSIZE,MSIZE),
5191 1 U(LSIZE,NSIZE,MSIZE),V(LSIZE,NSIZE,MSIZE),UE(NSIZE,MSIZE),
5192 2 VE(NSIZE,MSIZE),AV(LSIZE,NSIZE,MSIZE),ISIDE,AV0,AV00,WC(LSIZE),
5193 3 W(LSIZE,NSIZE,MSIZE),DV(LSIZE,NSIZE,MSIZE),DVO,DV00,CORGWS,
5194 4 RI(LSIZE,NSIZE,MSIZE),AH(NSIZE,MSIZE),AHC(NSIZE,MSIZE),THETSU(
5195 5 NSIZE,MSIZE),THETSV(NSIZE,MSIZE)
5196 C      EXTERNAL MODE VARIABLES
5197 COMMON/VEIS1/U(NSIZE,MSIZE),UHP(NSIZE,MSIZE),VH(NSIZE,MSIZE),
5198 1 VHP(NSIZE,MSIZE),D(NSIZE,MSIZE),SE(NSIZE,MSIZE),SEP(NSIZE,MSIZE),
5199 2 SEPP(NSIZE,MSIZE),FA(NPMSIZ),FB(NPMSIZ),GA(NPMSIZ),GB(NPMSIZ),
5200 3 ANB,ANC,SOLD(NSIZE,MSIZE),DHOST(NSIZE,MSIZE),VHOLD(NSIZE,MSIZE),
5201 4 PHI(NSIZE,MSIZE),TBX(NSIZE,MSIZE),TBY(NSIZE,MSIZE),ANA
5202 C      CONCENTRATION (SALINITY AND TEMPERATURE) VARIABLES
5203 COMMON/CONC1/KONCEN,ICFGO,ICOUP1,C1(LSIZE),RVERT,HRCOMC,
5204 1 S(LSIZE,NSIZE,MSIZE),MFUX(LSIZE,MSIZE),NEFLUX(NSIZE,MSIZE),
5205 2 GSTARX(NSIZE,MSIZE),GSTARZ(NSIZE,MSIZE),HRCON1,HRCON2,T(LSIZE,
5206 3 NSIZE,MSIZE),COST,CSOL1,CSOL2,CSOL3,SE,CSENS,CEVPI,CEVPZ,D10PCT,
5207 4 EPSLN,CTOT,OSUM,QA,Q1,QB,QS,QE,EA,ES,SOLAR,ALB,CWATER,CFAIR,ALV
5208 C      FILE HANDLING
5209 CHARACTER*40 FCON,FGEO,FINIT,FPRINT,FGRAPH,FMED
5210 COMMON/FILE1/FCON,FGEO,FINIT,FPRINT,FGRAPH,FMED,ISCR,LUKF,LUKIO,
5211 2 LUGRF,LUONU,LUMED,LUTID,LUWNU,LURIV,LUSAL,LUOCT,LUMET,LURVT

```